



NATIONAL AND KAPODISTRIAN UNIVERSITY OF ATHENS

**SCHOOL OF SCIENCES
DEPARTMENT OF INFORMATICS AND TELECOMMUNICATIONS**

PROGRAM OF POSTGRADUATE STUDIES

PhD THESIS

**Beyond Deep Learning: Enriching Data Representations
for Machine Learning Tasks**

Nikiforos I. Pittaras

ATHENS

October 2021



ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

**ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ**

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

**Πέρα από τη Βαθιά Μάθηση: Εμπλουτίζοντας
Αναπαραστάσεις Δεδομένων για Προβλήματα Μηχανικής
Μάθησης**

Νικηφόρος Ι. Πιτταράς

ΑΘΗΝΑ

Οκτώβριος 2021

PhD THESIS

Beyond Deep Learning: Enriching Data Representations for Machine Learning Tasks

Nikiforos I. Pittaras

SUPERVISOR: **Stamatopoulos Panagiotis**, Assistant Professor, Department of Informatics and Telecommunications, NKUA

THREE-MEMBER ADVISORY COMMITTEE:

Stamatopoulos Panagiotis, Assistant Professor, Department of Informatics and Telecommunications, NKUA

Emmanouil Koubarakis, Professor, Department of Informatics and Telecommunications, NKUA

Evangelos Karkaletsis, Research Director, NCSR “DEMOKRITOS”

SEVEN-MEMBER EXAMINATION COMMITTEE

Stamatopoulos Panagiotis,

Assistant Professor, Department of Informatics and Telecommunications, NKUA

Evangelos Karkaletsis,

Research Director, NCSR “DEMOKRITOS”

Ioannis Ioannidis,

Professor, Department of Informatics and Telecommunications, NKUA

Nikolaos Vasileiadis,

Professor, Department of Informatics, AUTH

Emmanouil Koubarakis,
Professor, Department of Informatics and Telecommunications, NKUA

Dimitrios Gunopulos,
Professor, Department of Informatics and Telecommunications, NKUA

Ion Androutsopoulos,
Professor, Department of Informatics, AUEB

Examination Date: August 31, 2021

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

Πέρα από τη Βαθιά Μάθηση: Εμπλουτίζοντας Αναπαραστάσεις Δεδομένων για
Προβλήματα Μηχανικής Μάθησης

Νικηφόρος Ι. Πιτταράς

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ: Σταματόπουλος Παναγιώτης, Επίκουρος Καθηγητής, Τμήμα Πληροφορικής και Τηλεπικοινωνιών, ΕΚΠΑ

ΤΡΙΜΕΛΗΣ ΕΠΙΤΡΟΠΗ ΠΑΡΑΚΟΛΟΥΘΗΣΗΣ:

Σταματόπουλος Παναγιώτης, Επίκουρος Καθηγητής, Τμήμα Πληροφορικής και Τηλεπικοινωνιών, ΕΚΠΑ

Εμμανουήλ Κουμπαράκης, Καθηγητής, Τμήμα Πληροφορικής και Τηλεπικοινωνιών, ΕΚΠΑ

Ευάγγελος Καρκαλέτσης, Διευθυντής Ερευνών, ΕΚΕΦΕ “ΔΗΜΟΚΡΙΤΟΣ”

ΕΠΤΑΜΕΛΗΣ ΕΞΕΤΑΣΤΙΚΗ ΕΠΙΤΡΟΠΗ

Σταματόπουλος Παναγιώτης,

Επίκουρος Καθηγητής, Τμήμα Πληροφορικής και Τηλεπικοινωνιών, ΕΚΠΑ

Ευάγγελος Καρκαλέτσης,

Διευθυντής Ερευνών, ΕΚΕΦΕ “ΔΗΜΟΚΡΙΤΟΣ”

Ιωάννης Ιωαννίδης,

Καθηγητής, Τμήμα Πληροφορικής και Τηλεπικοινωνιών, ΕΚΠΑ

Νικόλαος Βασιλειάδης,

Καθηγητής, Τμήμα Πληροφορικής, ΑΠΘ

Εμμανουήλ Κουμπαράκης,
Καθηγητής, Τμήμα Πληροφορικής και Τηλεπικοινωνιών,
ΕΚΠΑ

Δημήτριος Γουνόπουλος,
Καθηγητής, Τμήμα Πληροφορικής και Τηλεπικοινωνιών

Ιων Ανδρουτσόπουλος,
Καθηγητής, Τμήμα Πληροφορικής, ΟΠΑ

Ημερομηνία Εξέτασης: 31 Αυγούστου, 2021

ABSTRACT

This thesis conducts an investigation on data representation approaches for Machine Learning problems, focused on representation enrichment methods from knowledge resources. The study begins with a literature review on representations for classification over text, image and audio data, where methods were grouped to broad paradigms according to richness of information encompassed in the produced representation to a) low-level and template-matching approaches, b) aggregation-based methods and c) deep representation learning systems. After a comparison of pros and cons between paradigms, directions of potential improvements and extensions were identified, towards enhancing the richness of encapsulated information in the representation.

Subsequently, we moved on to specific proposals / extensions of representations for various learning problems, data modalities and domains, evaluated under novel applications and experimental evaluations. Specifically, different representations for text were evaluated for the Hate Speech Detection task on social media posts, the Automatic Summarization task for multiple domains (online articles, game reviews and social media texts), the Clustering / Event Detection task over articles and Social Media posts and the video classification task under a multimodal (image and audio) setting, over a variety of video datasets, labelling configuration and domain setting. This broad collection of studies on data representations verified the motivation of this thesis, namely that introduction of existing knowledge into representations is both under-utilized and a viable way of arriving at semantically rich features, for multiple representation extraction techniques.

Given this, we reiterated potential benefits of applying enrichment to Machine Learning problems and proceeded with a literature review of i) knowledge resources and ii) representation enrichment methods. This was conducted with respect to a classification task setting, considering text, images or audio data. We grouped enrichment approaches into three broad paradigms: a) input modification b) knowledge-guided representation refinement and c) end-to-end knowledge-aware systems. This comparative literature overview highlighted points of improvement and under-investigated areas, which led to adopting the approach of enriching deep neural content-based features with input modification methods. This is avenue pursued and investigated for the remainder of the thesis.

Given the above, two novel representation enrichment methods were proposed, with a focus on machine learning tasks for text data. First, we implemented a word embedding enrichment approach, using semantic information mined from the Wordnet knowledge resource. We investigated different techniques for data combination, knowledge extraction, diffusion and spread, dimensionality reduction and filtering and semantic disambiguation. We performed a large-scale experimental evaluation over multiple datasets and domains, along with statistical significance testing and a comparison to existing approaches. Our method was shown to the competition, with enrichment improving results significantly, enhancing prediction and representation explainability and yielding intuitive and predom-

inantly edge-case errors. Subsequently, the system was extended with different neural and conventional embeddings as well as proposed dimensionality reduction and clustering capabilities, all evaluated on the automatic summarization task over on encyclopedic articles.

Finally, we utilized the findings of this study into semantically enriched Hate Speech Detection system to be used in the Industry. The thesis is concluded by a summary of the totality of research work conducted, along with proposed directions of future study.

SUBJECT AREA: Machine Learning

KEYWORDS: Machine Learning, Neural Networks, Knowledge Resources

ΠΕΡΙΛΗΨΗ

Στην παρούσα μελέτη εξετάζονται αναπαραστάσεις δεδομένων για προβλήματα Μηχανικής Μάθησης, με έμφαση τον εμπλουτισμό τους με πληροφορία από πηγές γνώσεων.

Αρχικά, εκπονήθηκε βιβλιογραφική μελέτη για αναπαραστάσεις δεδομένων κειμένου, εικόνας και ήχου στο πρόβλημα της κατηγοριοποίησης. Έγινε συγκριτική καταγραφή και κατάταξη των μεθόδων σε α) αναπαραστάσεις χαμηλού επιπέδου και τοπικής εφαρμογής προτύπων β) συνδυασμός τοπικών χαρακτηριστικών με μεθόδους συνένωσης, συνδυασμού και μετασχηματισμού και γ) μοντέλα βαθιάς εκμάθησης αναπαραστάσεων. Έγινε μία σύγκριση θετικών και αρνητικών χαρακτηριστικών μεταξύ των τεχνικών και εντοπίστηκαν περιοχές βελτίωσης / επέκτασης τους για αναβάθμιση του σημασιολογικού περιεχομένου της παραγόμενης αναπαράστασης.

Στη συνέχεια, έγιναν ερευνητικές προτάσεις / επεκτάσεις μεθόδων αναπαράστασης σε διαφορετικά προβλήματα μηχανικής μάθησης και ποικίλων δεδομένων εισόδου σε στοχευμένες μελέτες και πειραματικές αξιολογήσεις. Συγκεκριμένα μελετήθηκαν διαφορετικές αναπαραστάσεις κειμένου για πρόβληματα όπως η Ανίχνευση Ρητορικής Μίσους σε δεδομένα από κοινωνικά δίκτυα και η Αυτόματη Εξαγωγή Περιλήψεων σε ποικιλία τύπου κειμένων (δημοσιογραφικά / εγκυκλοπαιδικά άρθρα, αξιολογήσεις ηλεκτρονικών παιχνιδιών, κείμενα σε ιστοσελίδες κοινωνικής δικτύωσης). Επιπλέον, έγινε μελέτη αναπαραστάσεων για Συσταδοποίηση / Εντοπισμό Γεγονότων σε κείμενο, καθώς και για την κατηγοριοποίηση βίντεο με αξιοποίηση αναπαράστασης εικόνας και ήχου. Το σύνολο της βιβλιογραφικής / ερευνητικής μελέτης ανέδειξε κατευθύνσεις βελτίωσης μεθόδων αναπαραστάσεων με τη χρήση υπάρχουσας πληροφορίας από δομημένες και υψηλής ποιότητας πηγές γνώσεων – τεχνική που είναι απούσα ή ελλιπής στη βιβλιογραφία.

Στη βάση αυτή, δόθηκε μία περιγραφή από πιθανά οφέλη που μπορεί να φέρει ο εμπλουτισμός με πληροφορία από εξωτερικές πηγές γνώσης. Επιπλέον, εκπονήθηκε βιβλιογραφική μελέτη με έμφαση σε μεθόδους εμπλουτισμού αναπαραστάσεων για διαφορετικούς τύπους δεδομένων (κείμενο, εικόνα και ήχος) και πηγών γνώσεων (οντολογίες, λεξικά, οπτικοακουστικές ιεραρχίες, κ.α.), για το πρόβλημα της ταξινόμησης. Επιπλέον, καταγράφηκαν λεπτομερώς υπάρχουσες μέθοδοι εμπλουτισμού και κατατάχθηκαν σε τρεις κατηγορίες: α) μέθοδοι εμπλουτισμού εισόδου με δεδομένα γνώσης β) μετασχηματισμός / συνδυασμός αναπαραστάσεων καθοδηγούμενος από γνώση και γ) συστήματα γνώσης βαθιάς μάθησης. Βάσει αυτής της μελέτης και αναγνωρίζοντας ελλείψεις και περιοχές βελτίωσης στην παρούσα βιβλιογραφία, προτάθηκε μία τεχνική εμπλουτισμού βασισμένη στον εμπλουτισμός εισόδου σε δεδομένα βαθιών αναπαραστάσεων, πάνω στην οποία επικεντρώθηκαν οι ερευνητικές προσπάθειες της διατριβής.

Με γνώμονα τα παραπάνω, μελετήθηκαν και προτάθηκαν δύο νέοι τρόποι εμπλουτισμού αναπαραστάσεων, δίνοντας έμφαση σε δεδομένα κειμένου. Αρχικά, αναπτύχθηκε ένα σύστημα νευρωνικών αναπαραστάσεων λέξεων, εμπλουτισμένων με σημασιολογική πληροφορία από την ιεραρχική οντολογία Wordnet. Ερευνήθηκαν διαφορετικοί τρόποι εμπλουτι-

σμού της εισόδου, τρόποι εξαγωγής σημασιολογίας από την οντολογία, τεχνικών διάχυσης βάρους στα δεδομένα γνώσης και προσεγγίσεων συνδυασμού της με τα χαρακτηριστικά περιεχομένου από το κείμενο. Έγινε πειραματική αξιολόγηση μεγάλης κλίμακας, ανάλυση στατιστικής σημαντικότητας και σύγκριση με άλλα συστήματα κατηγοριοποίησης και εμπλουτισμού, με χρήση μεγάλων συλλογών κειμένων ποικίλης θεματολογίας και χαρακτηριστικών. Η μέθοδος αποδίδει καλύτερα από υπάρχοντα συστήματα, και κατασκευάζει αναπαραστάσεις και μοντέλα μάθησης που είναι πιο αποδοτικά και παράγουν πιο εύκολα ερμηνεύσιμες προβλέψεις και χαρακτηριστικά. Στη συνέχεια, το παραπάνω σύστημα επεκτάθηκε με επιπλέον τεχνικές συμβατικών και νευρωνικών αναπαραστάσεων, διαφορετικές μεθόδους μείωσης διάστασης και τεχνικών συσταδοποίησης. Έγινε πειραματική αξιολόγηση στο πρόβλημα της αυτόματης εξαγωγής περιλήψεων σε δεδομένα από εγκυκλοπαιδικά άρθρα, η οποία επιβεβαίωσε τη συνεισφορά της προτεινόμενης μεθόδου εμπλουτισμού και ανέδειξε επιπλέον ενδιαφέροντα ευρήματα.

Τέλος, το σύνολο των ευρημάτων της μελέτης χρησιμοποιήθηκε για την κατασκευή ενός συστήματος εντοπισμού ρητορικής μίσους για αξιοποίηση στην βιομηχανία. Η παρούσα διατριβή κλείνει συνοψίζοντας το συνολικού ερευνητικό έργο και προσφέροντας κατευθύνσεις μελλοντικής επέκτασης της μελέτης που εκπονήθηκε.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Μηχανική Μάθηση

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: Μηχανική Μάθηση, Νευρωνικά Δίκτυα, Βάσεις Γνώσεων

ACKNOWLEDGEMENTS

I would like to thank my advisors, Prof. P. Stamatopoulos, Prof. M. Koubarakis and Dr. E. Karkaletsis for their help, guidance, and support to the completion of this thesis.

I would also like to acknowledge Dr. G. Giannakopoulos, who provided invaluable guidance and mentoring, helped keep things into perspective and nudged things forward when needed.

Finally, I am grateful to my parents for their mysterious faith in my pursuits, and of course, to C. Themeli for her constant encouragement, unwavering patience and dependable assistance during the past four years.

Last but not least, this thesis has been financially supported by the Stavros Niarchos Foundation (SNF)¹ and Athens Technology Center², through the SNF Industrial Ph.D. Scholarships programme.

¹<https://www.snfcc.org/>

²<https://www.atc.gr/>

LIST OF PUBLICATIONS

1. N. Pittaras, G. Giannakopoulos, G. Papadakis, V. Karkaletsis, "Text classification with semantically enriched word embeddings", Natural Language Engineering Special Issue: Informing Neural Architectures for NLP with Linguistic and Background Knowledge
2. N. Pittaras, V. Karkaletsis, "A study of semantic augmentation of word embeddings for extractive summarization", Multiling Workshop, RANLP2019, Varna, Bulgaria (proceedings).
3. N. Gialitsis, N. Pittaras, P. Stamatopoulos, "A topic-based sentence representation for extractive text summarization", Multiling Workshop, RANLP2019, Varna, Bulgaria.
4. C. Themeli, G. Giannakopoulos, N. Pittaras "A study of text representations for Hate Speech Detection", CICLING 2019, La Rochelle, France.
5. N. Pittaras, G. Papadakis, G. Stamoulis, G. Argyriou, E. K. Taniskidou, E. Thanos, G. Giannakopoulos, L. Tsekouras, E. Koubarakis, "GeoSensor: Semantifying Change and Event Detection over Big Data", SAC 2019, Limassol, Cyprus.
6. A. Kosmopoulos, A., Liapis, A., Giannakopoulos, G., Pittaras, N.. Summarizing Game Reviews: First Contact. SETN Workshops, 2020.
7. Giannakopoulos, G., Kiomourtzis, G., Pittaras, N., Karkaletsis, V. (2020). Scaling and Semantically-Enriching Language-Agnostic Summarization. In A. Fiori (Ed.), Trends and Applications of Text Summarization Techniques (pp. 244-292). IGI Global.

ΣΥΝΟΠΤΙΚΗ ΠΑΡΟΥΣΙΑΣΗ ΤΗΣ ΔΙΔΑΚΤΟΡΙΚΗΣ ΔΙΑΤΡΙΒΗΣ

Εισαγωγή

Η μελέτη αυτή εστιάζει σε τεχνικές Μηχανικής Μάθησης που αποτελούν καίριο συστατικό ψηφιακών εφαρμογών στην αγορά, έρευνα και βιομηχανία, και εκμεταλλεύονται τεράστιες ποσότητες δεδομένων κειμένου και πολυμέσων. Η αποδοτική αναπαράσταση τέτοιων δεδομένων σε ένα σύστημα Μάθησης είναι καθοριστικής σημασίας για την ορθότητα και ερμηνευσιμότητα (explainability) των παραγόμενων αποτελεσμάτων. Στη τρέχουσα διατριβή, μελετώνται μέθοδοι εμπλουτισμού αναπαραστάσεων με υπάρχουσα, δομημένη πληροφορία από οντολογίες, βάσεις γνώσεων και αποθετήρια δομημένης γνώσης υψηλής ποιότητας. Η διερεύνηση αυτή στοχεύει να αντιμετωπίσει υπάρχουσες αδυναμίες και ελλείψεις τεχνικών στη βιβλιογραφία και να παράξει σημασιολογικά πλούσιες, ερμηνεύσιμες αναπαραστάσεις που παράγουν συστήματα Μηχανικής Μάθησης τελευταίας τεχνολογίας (state of the art) και ανταγωνιστικής απόδοσης.

Υιοθετούμε τον παραπάνω στόχο αυτό στα πλαίσια ενός ευρέους αντικείμενου μελέτης, ερευνώντας διαφορετικά προβλήματα μάθησης, κατηγορίες αναπαράστασης και τύπους δεδομένων εισόδου. Για αυτό τον λόγο, δίνεται ένας γενικός ορισμός ενός προβλήματος Μηχανικής Μάθησης και περιγράφονται τυπικά δομικά συστατικά ενός τέτοιου συστήματος, συχνά οργανωμένα σε μία αλυσίδα ή μονοπάτι υπολογιστικών βημάτων (pipeline). Στη δομή αυτή, δίνεται έμφαση στη σημασία του βήματος της εξαγωγής αναπαράστασης: αποτελεί κομβικό σημείο απώλειας πληροφορίας που εμπεριέχεται σε αντικείμενα πραγματικού κόσμου της εισόδου. Αυτή η απώλεια αποτελεί πηγή θορύβου και/ή λάθους, που διαιωνίζεται σε όλους τους υπολογισμούς που ακολουθούν και επηρεάζουν την απόδοση του συστήματος. Ως πιθανή λύση, προτείνεται ο εμπλουτισμός των αναπαραστάσεων με πληροφορία και γνώση από σχετικές εξωτερικές πηγές, προσέγγιση που αναλύεται στα επόμενα κεφάλαια.

Αναπαραστάσεις Βασισμένες στο Περιεχόμενο

Η ανάλυση ξεκινά με μία εκτενή μελέτη της βιβλιογραφίας σχετική με την κατασκευή αναπαραστάσεων δεδομένων, αρχικά δίνοντας έμφαση σε αναπαραστάσεις που πηγάζουν αποκλειστικά από το περιεχόμενο (content) – χωρίς αξιοποίηση εξώτερης γνώσης. Εστιάζοντας στο πρόβλημα της κατηγοριοποίησης σε δεδομένα ποικίλης μορφής (κειμένο, εικόνα ή ήχος), καταγράφηκαν μέθοδοι εξαγωγή αναπαραστάσεων που οργανώθηκαν σε τρεις κατηγορίες:

1. **Μέθοδοι χαμηλού επιπέδου και τοπικής εφαρμογής προτύπων:** εδώ, ένα σύνολο από προκαθορισμένα μοτίβα που ορίζουν απλούς υπολογισμούς εφαρμόζεται

τοπικά ή ολικά στην είσοδο, παράγοντας διακριτές αποκρίσεις που χρησιμοποιούνται ως χαρακτηριστικά. Τεχνικές όπως Bag of Words, περιγραφείς (descriptors) οπτικού / ακουστικού περιεχομένου και απλές στατιστικές / μετρικές (π.χ. SNR) ανήκουν σε αυτή την ομάδα εφαρμογών.

2. **Τεχνικές συνδυασμού τοπικών χαρακτηριστικών:** μέλη της κατηγορίας αυτής χρησιμοποιούν χαρακτηριστικά της προηγούμενης στρατηγικής, συγχωνεύοντάς, μετασχηματίζοντας, συνενώνοντας και συνδυάζοντάς τα με προκαθορισμένες τεχνικές σε πιο αποδοτικές αναπαραστάσεις βελτιωμένης εκφραστικότητας. Για παράδειγμα, μέθοδοι μετασχηματισμού / παραγοντοποίησης πινάκων (π.χ. SVD, LSA), συσταδοποίησης (π.χ. K-Means) και μοντελοποίησης θέματος.
3. **Μοντέλα βαθιάς εκμάθησης αναπαραστάσεων:** Εδώ παράγονται ιεραρχίες χαρακτηριστικών (από χαμηλού επιπέδου σε αφηρημένα, πλούσια features) με τεχνικές εκμάθησης αναπαραστάσεων (representation learning). Συχνά, τέτοια μοντέλα χαρακτηρίζονται με δυισμό παράλληλης μάθησης της αναπαράστασης και του προβλήματος. Παρατηρείται υψηλή δυνατότητα για επαναχρησιμοποίηση κατασκευασμένων αναπαραστάσεων σε διαφορετικά προβλήματα και κατηγορίες δεδομένων με σχετικά μικρή απώλεια στην απόδοση. Νευρωνικές τεχνικές για εκμάθηση διανυσμάτων λέξεων (Word Embeddings), δίκτυα συνέλιξης (Convolutional Neural Networks) και αναδρομικού / ακολουθιακού υπολογισμού (Recurrent Neural Networks) είναι μέλη αυτής της στρατηγικής.

Οι κατηγορίες αυτές παράγουν αναπαραστάσεις που χαρακτηρίστηκαν ως χαμηλού, μέσου και υψηλού επιπέδου αντίστοιχα, αναφορικά με το πόσο πλούσια είναι η πληροφορία που εμπεριέχουν, δηλαδή με το βαθμό που η αναπαράσταση μπορεί να αντικατοπτρίζει έννοιες ποικίλης σημασιολογίας, αφαιρετικότητας και ποσότητας πληροφορίας. Για κάθε εργασία στην βιβλιογραφία που μελετήθηκε, έγινε συγκριτική καταγραφή της μεθόδου αναπαράστασης, μοντέλου μάθησης, μετρικών απόδοσης και τύπου προβλήματος κατηγοριοποίησης (π.χ. δυαδικής ταξινόμησης / πολλών κλάσεων / πολλαπλών ετικετών ανά δεδομένο). Ακολούθησε μία σύγκριση των κατηγοριών σχετικά με τον σημασιολογικό πλούτο που παράγουν στην έξοδό τους, στο βαθμό επεξηγησιμότητας τους, με το αν βασίζονται σε μάθηση ή σε στατικό προκαθορισμό της αναπαράστασης, στη διάσταση της διανυσματικής τους εξόδου, καθώς και στις ανάγκες τους σε ποσότητας δεδομένων και/ή υπολογιστικής ισχύος. Επιπλέον, εντοπίστηκαν περιοχές βελτίωσης / επέκτασης τέτοιων μεθόδων για κατασκευή αναπαραστάσεων με πλούσια σημασιολογία και υψηλής ποιότητας χρήσιμης πληροφορίας.

Καινοτόμες Μελέτες και Εφαρμογές Αναπαραστάσεων Περιεχομένου

Με την ολοκλήρωση της βιβλιογραφικής ανασκόπησης, έγινε μία σειρά στοχευμένων μελετών για καινοτόμες προτάσεις και επεκτάσεις μεθόδων και εφαρμογών αναπαραστάσεων περιεχομένου, σε διαφορετικά προβλήματα μηχανικής μάθησης και ποικίλων δεδομένων εισόδου. Στα πλαίσια αυτών των εργασιών έγιναν ολοκληρωμένες μελέτες, δημοσιεύσεις

και πειραματικές αξιολογήσεις μεγάλης κλίμακας. Προβλήματα που εξετάστηκαν συμπεριλαμβάνουν:

- **Ανίχνευση Ρητορικής Μίσους:** Αρχικά, μελετήθηκαν διαφορετικές αναπαραστάσεις κειμένου για το πρόβλημα της Ανίχνευση Ρητορικής Μίσους σε δεδομένα από κοινωνικά δίκτυα όπως το Twitter. Αξιολογήθηκαν διαφορετικές αναπαραστάσεις όπως γράφοι n-gram graphs, Bag of Words και Word Embeddings, καθώς και διαφορετικοί ταξινομητές – για παράδειγμα, K κοντινότεροι γείτονες (K-Nearest Neighbors), νευρωνικά δίκτυα και Random Forests.
- **Αυτόματη Εξαγωγή Περιλήψεων:** Στη συνέχεια έγινε έρευνα σε αναπαραστάσεις για αυτόματη εξαγωγή περιλήψεων σε εγκυκλοπαιδικά κείμενα. Εξετάστηκε η απόδοση αναπαραστάσεων βασισμένες στην μοντελοποίηση θέματος (Topic Modeling) για επιλογή ή μη προτάσεων ως μέρη της τελικής περιληψης (τεχνική Extractive Summarization) με γραμμικούς και μη ταξινομητές, σε δεδομένα δημοσιογραφικών και εγκυκλοπαιδικών άρθρων στο διαδίκτυο. Επεκτείναμε την έρευνα στην εξαγωγή περιλήψεων στο καινοτόμο πεδίο των αξιολογήσεων (reviews) ηλεκτρονικών παιχνιδιών σε διαδικτυακές πλατφόρμες, εξετάζοντας μεθόδους εξαγωγής πληροφορίας και σημείων ενδιαφέροντος (aspects) μίας αξιολόγησης μέσω συσταδοποίησης, καθώς και διαφορετικές αναπαραστάσεις (π.χ. από βαθιά νευρωνικά δίκτυα αρχιτεκτονικής BERT).
- **Εξαγωγή Περιλήψεων, Κατηγοριοποίηση και Συσταδοποίηση:** Έπειτα, έγινε μία ευρεία μελέτη πάνω σε αναπαραστάσεις γράφων n-gram graphs για προβλήματα συσταδοποίησης (clustering), ανίχνευσης γεγονότων, ταξινόμησης και εξαγωγής περιληψης σε δεδομένα κειμένου από άρθρα και ιστοσελίδες κοινωνικής δικτύωσης. Επιπλέον, μελετήθηκαν και αξιολογήθηκαν τρόποι κλιμάκωσης (scaling) των συστημάτων παραγωγής και σύγκρισης αναπαραστάσεων για μεγάλα δεδομένα (big data) μέσω κατανεμημένης εκτέλεσης με το SPARK framework.
- **Κατηγοριοποίηση βίντεο:** Πέραν από δεδομένα κειμένου, έγινε διερεύνηση σε πολυτροπικές (multimodal) τεχνικές νευρωνικών αναπαραστάσεων βίντεο ως συνδυασμό εικόνας και ήχου, για το πρόβλημα της κατηγοριοποίησης. Έγινε αξιολόγηση διαφορετικών υπολογιστικών ροών και αρχιτεκτονικών, με εφαρμογή σε ποικιλία από τύπους ετικετών, θεματολογίας και ποιότητας βίντεο καθώς και μεγέθους συλλογών δεδομένων. Εφαρμόστηκαν αρχιτεκτονικές νευρωνικής επεξεργασίας μέσω συνέλιξης με δίκτυα CNN (Convolutional Neural Networks) καθώς και επεξεργασίας ακολουθιών με LSTM (Long Short Term Memory) τοπολογίες.

Με το πέρας της βιβλιογραφικής και ερευνητικής μελέτης των μεθόδων αναπαραστάσεων, επιβεβαιώθηκε το κίνητρο και ο στόχος της διατριβής: η χρήση υπάρχουσας πληροφορίας από δομημένες και υψηλής ποιότητας πηγές γνώσεων είναι απούσα ή ελλιπής στη βιβλιογραφία, με την πλειοψηφία των τεχνικών αναπαράστασης να επικεντρώνεται στο περιεχόμενο των δεδομένων ενός παραδείγματος. Επιπλέον, δεν αναγνωρίστηκαν καθολικά ανώτερες αναπαραστάσεις περιεχομένου για κάθε πρόβλημα και θεματική περιοχή. Τέλος, οι εφαρμογές και μελέτες ανέδειξαν τρόπους και τεχνικές εισαγωγής πληροφορίας γνώσης σε πληθώρα αναπαραστάσεων.

Τεχνικές και Πηγές Εμπλουτισμού Αναπαραστάσεων

Στο σημείο αυτό η μελέτη στράφηκε στον εμπλουτισμό αναπαραστάσεων από πηγές εξωτερικής πληροφορίας.

Αρχικά, δόθηκε περαιτέρω έμφαση στα πιθανά οφέλη που μπορεί να επιφέρει ο εμπλουτισμός συστημάτων Μηχανικής Μάθησης με πηγές γνώσεων. Αυτά συμπεριλαμβάνουν την προσθήκη πληροφορίας για συμφραζόμενα, επίλυση αμφισημίας στο σύνολο δεδομένων ή στη διαδικασία παραγωγής τους, καθώς και τεχνικούς / ειδικούς όρους. Επιπλέον, ο εμπλουτισμός μπορεί να ανταποκριθεί σε ανάγκες για ερμηνευσιμότητα αναπαραστάσεων και συστημάτων μάθησης, σε ελλείψεις δεδομένων (για παράδειγμα, σε δεδομένα κειμένου για γλώσσες με λίγους πόρους για κατασκευή μοντέλων), και στις τεράστιες ανάγκες για υπολογιστική ισχύ και ενέργεια (όπου απαιτούνται μεγάλες ποσότητες ενέργειας για εκπαίδευση πολύ μεγάλων μοντέλων σε ειδικευμένο υλικό, για παράδειγμα κάρτες γραφικών για βαθιά μάθηση). Λόγω των παραπάνω, η παρούσα διατριβή στοχεύει στη μελέτη τεχνικών εμπλουτισμού αναπαραστάσεων για συστήματα Μηχανικής Μάθησης.

Αρχικά, εκπονήθηκε ανασκόπηση της υπάρχουσας βιβλιογραφίας σχετικής με μεθόδους εμπλουτισμού αναπαραστάσεων για διαφορετικούς τύπους δεδομένων (κείμενο, εικόνα και ήχος) και πηγών γνώσεων (οντολογίες, λεξικά, οπτικοακουστικές ιεραρχίες, κ.α.). Δόθηκε έμφαση στο πρόβλημα της κατηγοριοποίησης, και μέθοδοι στην βιβλιογραφία κατατάχθηκαν σε τρείς κατηγορίες, ως προς τον τρόπο εμπλουτισμού αναπαράστασης. Αυτές είναι:

- Μέθοδοι εμπλουτισμού εισόδου**, όπου πληροφορία από εξωτερικές πηγές γνώσης συνδυάζεται στην αναπαράσταση περιεχομένου όπου παραμένει ως διακριτό συστατικό, προσφέροντας εξηγήσιμα και σημασιολογικά πλούσια χαρακτηριστικά.
- Μετασχηματισμός αναπαράστασης καθοδηγούμενος από γνώση**: εδώ, αξιοποιούνται πηγές γνώσης για να κατευθύνουν, ενημερώσουν και επηρεάσουν προκαθορισμένους τρόπους συνδυασμού και/ή μετασχηματισμού αναπαραστάσεων περιεχομένου, καταλήγοντας σε υβριδικές, εμπλουτισμένες αναπαραστάσεις μειωμένης διάστασης και αυξημένης εκφραστικότητας.
- Συστήματα γνώσης βαθιάς μάθησης**: τέτοιες τεχνικές χρησιμοποιούν βαθιά νευρωνικά δίκτυα για ταυτόχρονη μοντελοποίηση του προβλήματος, της εκμάθησης αναπαραστάσεων και του συνδυασμού πληροφορίας περιεχομένου και γνώσης, παρουσιάζοντας υψηλές δυνατότητες για μεταφορά μάθησης (transfer learning) και παράγοντας πολλαπλές ιεραρχίες εμπλουτισμένων χαρακτηριστικών.

Έγινε συγκριτική παρουσίαση της βιβλιογραφίας που εμπίπτει στις παραπάνω κατηγορίες, καταγράφοντας τον τρόπο εμπλουτισμού, την πηγή γνώσης και λεπτομέρειες ως προς την αναπαράσταση, μάθηση, αξιολόγηση και τύπο προβλήματος κατηγοριοποίησης. Παρατηρήθηκαν ομοιότητες μεταξύ των κατηγοριών αναπαραστάσεων περιεχομένου, με παρόμοια πλεονεκτήματα, μειονεκτήματα και ελλείψεις να ισχύουν και για τις κατηγορίες εμπλουτισμού.

Με τη θεώρηση της σχετικής έρευνας αναγνωρίστηκε μία περιοχή με ελλιπή αξιολόγηση / μελέτη από υπάρχουσες εργασίες όπου αποφασίστηκε η πραιτέρω διερεύνηση, και παρουσιάζει σημαντικά οφέλη. Συγκεκριμένα, επιλέχθηκε η ερευνητική κατεύθυνση εμπλουτισμού με χρήση:

- **Εμπλουτισμού εισόδου**, όπου χρησιμοποιείται εξωτερική πληροφορία για εμπλουτισμό της αναπαράστασης με διακριτό τρόπο. Η χρήση αυτής της στρατηγικής επιτρέπει στα χαρακτηριστικά που προέρχονται από πηγές γνώσης να εντοπιστούν στην μικτή αναπαράσταση και διατηρήσουν την σημασιολογία τους, ενισχύοντας την εκφραστικότητα με τρόπο που είναι διαφανής και επεξηγήσιμος από τον άνθρωπο.
- **Εμπλουτισμό βαθιών αναπαραστάσεων**, που αποτελούν ισχυρές τεχνικές εκμάθησης αναπαραστάσεων με δυνατότητα να εξάγουν χρήσιμη πληροφορία από πολύ μεγάλες ποσότητες δεδομένων.

Αυτός ο συνδυασμός εκμεταλλεύεται τα πλεονεκτήματα των επιλεγμένων κατηγοριών αναπαραστάσεων περιεχομένου και εμπλουτισμού. Ταυτόχρονα, η καθεμία να επιδιώκει να αντιμετωπίσει τις ελλείψεις της άλλης. Αυτή η ερευνητική κατεύθυνση μελετάται στα επόμενα κεφάλαια.

Προτάσεις Εμπλουτισμού Αναπαραστάσεων Βαθιάς Μάθησης

Με γνώμονα τις παραπάνω παρατηρήσεις, προτάθηκε και μελετήθηκε μία καινοτόμος τεχνική για εμπλουτισμό αναπαράστασης: η βελτίωση μεθόδων βαθιών αναπαραστάσεων από νευρωνικά μοντέλα, με πληροφορία από πηγές γνώσης. Δίνοντας έμφαση σε δεδομένα κειμένου, η πρόταση υλοποιήθηκε μέσω ενός σύστημα εμπλουτισμού νευρωνικών διανυσματικών αναπαραστάσεων λέξεων, με σημασιολογική πληροφορία από την ιεραρχική οντολογία Wordnet. Η οντολογία αυτή αποτελεί έναν γράφο όπου κόμβοι αντιστοιχούν σε έννοιες, και ακμές υποδηλώνουν σχέσεις μεταξύ τους (π.χ. σχέση υπωνυμίας, μερωνυμίας, κ.α.). Η δομή αυτή χρησιμοποιήθηκε για αντιστοίχιση λέξεων του κειμένου με σύνολα σχετικής εννοιολογικής πληροφορίας, εμπλουτίζοντας έτσι το κείμενο με πλούσια σημασιολογία υψηλής ποιότητας.

Συνολικά, ερευνήθηκε πληθώρα διαφορετικών τρόπων και παραμέτρων για εμπλουτισμό της εισόδου, όπως:

- Τρόποι αποσαφήνισης σημασιολογίας, δεδομένης λέξης εισόδου, έτσι ώστε να ταυτοποιείται σωστά η έννοια μίας πιθανώς αμφίσημης λέξης, για εξαγωγή σωστής πληροφορίας από την πηγή γνώσης: εξετάστηκαν διάφορες υλοποιήσεις, από χρήση της πιο δημοφιλούς έννοιας για μία λέξη, εώς προσεγγίσεις κατασκευής και σύγκρισης διανυσματικών αναπαραστάσεων των σημασιολογικών κόμβων στο Wordnet
- Τεχνικές συγκερασμού δεδομένων γνώσης και περιεχομένου. Συγκεκριμένα, δοκιμάστηκε η συνένωση (concatenation) των διανυσμάτων περιεχομένου και γνώσης, καθώς και η χρήση μόνο του τελευταίου για την αναπαράσταση.

- Μέθοδοι διάχυσης και ελέγχου της ποσότητας πληροφορίας από τον σημασιολογικό γράφο. Εδώ αξιοποιήθηκε η δομή της οντολογίας Wordnet: εκτελέστηκε ένα πολυβηματικό μονοπάτι κατά μήκος της ακμής υπερυψύματος από την αρχική έννοια, για άντληση όλο και γενικότερων εννοιών σχετικών με τη λέξη εισόδου. Στις επιπλέον ενεργοποιήσεις αυτές ανατέθηκε μειωμένο βάρος στην αναπαράσταση, όσο πιο μακριά βρίσκονται από τον αρχικό κόμβο.
- Μεθοδολογίες περιορισμού διάστασης της σημασιολογικής αναπαράστασης. Για αντιμετώπιση της “κατάρας της διαστατικότητας”, εφαρμόστηκαν περιορισμοί στο πλήθος των εννοιών που χρησιμοποιήθηκαν συναρτήσει των συχνοτήτων εμφάνισης εννοιών, όπως η εφαρμογή κατωφλίων ή η διατήρηση των πιο δημοφιλών εννοιών.

Με βάση τα παραπάνω, έγινε πειραματική αξιολόγηση μεγάλης κλίμακας, ανάλυση στατιστικής σημαντικότητας και σύγκριση με άλλα συστήματα κατηγοριοποίησης και εμπλουτισμού. Χρησιμοποιήθηκε νευρωνικός ταξινομητής με εκπαίδευση σε cross-validation και early stopping σχηματισμούς, σε δύο datasets διαφορετικού μεγέθους, θεματολογίας (π.χ. από κείμενα γενικής φύσεως, μέχρι ειδησεογραφικά άρθρα και ιατρικές επιστημονικές αναφορές), πλήθους από ετικέτες και γλωσσολογικά / λεξιλογικά χαρακτηριστικά. Η αξιολόγηση έδειξε πως η προτεινόμενη μέθοδος οδηγεί σε στατιστικά σημαντικές βελτιώσεις στην απόδοση της ταξινόμησης, ανέδειξε τον βέλτιστο συνδυασμό παραμέτρων, αρχιτεκτονικών επιλογών και οδών επέκτασης και αναβάθμισης της ερευνητικής μας πρότασης.

Περαιτέρω αξιολογήσεις έδειξαν πως η προτεινόμενη μέθοδος αποδίδει καλύτερα από σύγχρονες σχετικές τεχνικές βαθιών νευρωνικών αναπαραστάσεων λέξεων και κειμένου, καθώς και από άλλες σύγχρονες τεχνικές εμπλουτισμού. Παρόμοια αποτελέσματα πάρθηκαν από επέκταση της έρευνας σε επιπλέον σύνολα δεδομένων διαφορετικών θεματολογιών (όπως επιστημονικές δημοσιεύσεις ιατρικού τομέα και ειδησεογραφικά κείμενα). Λεπτομερής ανάλυση των λανθασμένων προβλέψεων του συστήματός μας έδειξε πως οι ταξινομητές εμπλουτισμένης πληροφορίας που προτάθηκαν δίνουν λανθασμένες προβλέψεις σε αμφίσημες περιπτώσεις δεδομένων ή ετικετών, αναδεικνύοντας την ευκολία κατανόησης του συστήματός μας. Η ερμηνευσιμότητα ενισχύεται επιπλέον με τη δυνατότητα ελέγχου και ανάλυσης των παραγόμενων εμπλουτισμένων αναπαραστάσεων, που περιέχουν κατανοητά και καλά ορισμένα συστατικά στοιχεία.

Στη συνέχεια, υλοποιήσαμε επέκταση του συστήματος για μελέτη εμπλουτισμού επιπλέον – νευρωνικών και μη – αναπαραστάσεων και έλεγχο επιπλέον μεθόδων μείωσης διάστασης, όπως μετασχηματισμό πινάκων και τεχνικών συσταδοποίησης. Έγινε αξιολόγηση στο πρόβλημα της εξαγωγής περιλήψεων σε επίπεδο επιλογής πρότασης, ως εφαρμοσμένη κατηγοριοποίηση μικρών κειμένων, όπου ζητείται η δυαδική κατηγοριοποίηση προτάσεων ως μέλη της τελικής περίληψης που θα παραχθεί.

Εφαρμόστηκε το προτεινόμενο σύστημα με εμπλουτισμό από την οντολογία Wordnet, με συνένωση του σημασιολογικού διανύσματος με αναπαραστάσεις βασισμένες σε word embeddings και στην τεχνική TF-IDF. Ως ταξινομητής υλοποιήθηκαν αρχιτεκτονικές πολυεπίπεδων νευρωνικών δικτύων, ενώ εφαρμόστηκαν τεχνικές δειγματοληψίας για ανακατασκευή του συνόλου δεδομένων σε μορφή που να υποστηρίζει αποδοτική εκπαίδευση.

Τα αποτελέσματα αξιολόγησης σε εγκυκλοπαιδικά άρθρα στο διαδίκτυο με δεδομένες περιλήψεις κατασκευασμένες από ανθρώπους επιβεβαίωσαν πως η προτεινόμενη μέθοδος αυξάνει την απόδοση στο νέο πρόβλημα, σε σύγκριση με χρήση των αναπαραστάσεων περιεχομένου που εξετάστηκαν. Επιπλέον, αναδείχθηκαν προοπτικές επιπλέον βελτίωσης με μείωση διάστασης της αναπαράστασης, με τη μέθοδο μετασχηματισμού LSA να δίνει την μεγαλύτερη αύξηση απόδοσης.

Στο τέλος του κεφαλαίου παρουσιάστηκε μία περίληψη των ερευνητικών αποτελεσμάτων και συμπερασμάτων του έργου της διατριβής σχετικά με τον εμπλοουτισμό αναπαραστάσεων.

Εφαρμογή στη Βιομηχανία

Η παρούσα διδακτορική διατριβή κατατέθηκε ως πρόταση και κέρδισε υποτροφία Βιομηχανικού Διδακτορικού από το Ίδρυμα Σταύρος Νιάρχος³, με μερική συμμετοχή και υποστήριξη από εταιρεία του βιομηχανικού τομέα. Ως αποτέλεσμα, τα ευρήματα από το σύνολο των βιβλιογραφικών μελετών, των στοχευμένων εφαρμογών αναπαραστάσεων και των μεθόδων εμπλοουτισμού χρησιμοποιήθηκαν για την κατασκευή μίας εφαρμογής βιομηχανικών προδιαγραφών, προς αξιοποίηση από την συμμετέχουσα εταιρεία.

Για το σκοπό αυτό, επικοινωνήθηκαν και καταγράφηκαν οι επιχειρησιακές και τεχνικές ανάγκες, και αποφασίστηκε η κατασκευή ενός συστήματος ανίχνευσης και ταξινόμησης ρητορικής μίσους (Hate Speech Detection) σε δεδομένα κειμένου. Αρχικά, συλλέχθηκε ένα σύνολο δεδομένων με ποικίλες τεχνικές, όπως ενσωμάτωση υπαρχόντων datasets και συλλογή κειμένων από τον ιστό (crawling), καταλήγοντας σε δείγματα από 5 κατηγορίες μίσους: ρατσισμού, σεξισμού, σεξουαλικού προσανατολισμού, θρησκευτικού σωβινισμού και μία αρνητική κλάση. Πάνω σε αυτό το dataset κατασκευάστηκε ένα αρθρωτό σύστημα, με δομικά συστατικά μέρη όπως

- Λειτουργίες επεξεργασίας και φιλτραρίσματος κειμένου συγκεκριμένης θεματολογίας / πηγής
- Υλοποίηση νευρωνικών και συμβατικών μεθόδων αναπαράστασης περιεχομένου για δεδομένα κειμένου
- Μέθοδοι αναπαράστασης γνώσης, ενσωματώνοντας πηγές πληροφορίας γενικού (π.χ. Wordnet) αλλά και ειδικού (χρήσιμες κυρίως σε προβλήματα ανίχνευσης ρητορικής μίσους) χαρακτήρα
- Εμπλοουτισμό αναπαράστασης, όπου εφαρμόζονται οι βέλτιστες κατευθύνσεις, όπως αναδείχθηκαν από τα ευρήματα της διατριβής
- Μοντέλα μάθησης, όπου υποστηρίχθηκαν, μεταξύ άλλων, τεχνικές νευρωνικών ταξινομητών και λογιστικής παλινδρόμησης

³<https://www.snfcc.org/>

- Πολλαπλές μετρικές αξιολόγησης κατηγοριοποίησης, τεχνικές εκπαίδευσης (π.χ. μέθοδοι cross-validation) και αποθήκευσης των αποτελεσμάτων εκπαίδευσης και αξιολόγησης

Επιπλέον, ενσωματώθηκαν στο σύστημα δημοφιλή εργαλεία διαχείρισης για ML Pipelines. Αυτά περιλαμβάνουν εργαλεία για παρακολούθηση και σύγκριση απόδοσης μεμονωμένων και διαφορετικών μοντέλων, βελτιστοποίηση παραμέτρων, διάθεση μοντέλων για εφαρμογή (deployment) σε RESTful API σχηματισμούς και εκπαίδευσης με διαφορετικούς συνδυασμούς και μέρη του συνόλου δεδομένων. Πραγματοποιήθηκε υλοποίηση και βελτιστοποίηση του κώδικα σε πολλαπλές εκδόσεις, εκπαίδευση υποψήφιων μοντέλων, έλεγχος αρχιτεκτονικών, υπερπαραμέτρων, τεχνικών αξιολόγησης και συνόλων δεδομένων. Το πειραματικά βέλτιστο μοντέλο, τα αποτελέσματα, και οι τελικές εκδόσεις όλων των απαραίτητων δομικών συστατικών, εργαλείων, δεδομένων και σχετικών πόρων παραδόθηκαν στη συνεργαζόμενη εταιρεία.

Επίλογος

Η διατριβή κλείνει με μία σύνοψη του συνολικού ερευνητικού έργου και συνεισφορών που παρήχθησαν στα πλαίσια της μελέτης που διεξήχθη. Αυτές συμπεριλαμβάνουν α) την μελέτη βιβλιογραφίας για αναπαραστάσεις περιεχομένου, β) καινοτόμες προτάσεις και εφαρμογές αναπαραστάσεων σε διαφορετικά προβλήματα μάθησης και δεδομένα εισόδου, γ) την βιβλιογραφική μελέτη τεχνικών εμπλουτισμού και πηγών γνώσεων, δ) την πρόταση καινοτόμων συστημάτων εμπλουτισμού αναπαραστάσεων, αξιολόγησή τους σε διαφορετικά προβλήματα και σύνολα δεδομένων, και επιβεβαίωση της αποτελεσματικότητάς της προτεινόμενης προσέγγισης, και ε) την υλοποίηση, βελτιστοποίηση και εφαρμογή συστήματος Ανίχνευσης Ρητορικής Μίσους για αξιοποίηση στη βιομηχανία, χρησιμοποιώντας σύγχρονες πρακτικές, εργαλεία και τεχνικές και παραδίδοντας εκπαιδευμένα μοντέλα και σύνολα δεδομένων στη συνεργαζόμενη με το έργο εταιρεία.

Τέλος, προσφέρονται κατευθύνσεις επέκτασης της παρούσας εργασίας, όπως η αξιοποίηση διαφορετικών / πολλαπλών πηγών γνώσεων, η διερεύνηση επιπλέον μεθόδων μείωσης διάστασης (π.χ. autoencoders), η επέκταση των διαθέσιμων πηγών γνώσεων για δεδομένα κειμένου / ήχου, ο συνδυασμός πολλαπλών τεχνικών εμπλουτισμού και η ανάπτυξη εργαλείων για βελτίωση της επεξηγησιμότητας (explainability) της προτεινόμενης προσέγγισης.

CONTENTS

1 INTRODUCTION	37
1.1 Machine Learning in the Modern Digital Landscape	37
1.2 Structure of a Machine Learning System	37
1.3 Enriching Data Representations	39
1.4 Contributions	40
1.5 Thesis structure	41
2 CONTENT-BASED REPRESENTATION APPROACHES: AN OVERVIEW	43
2.1 Categorizing content-based approaches	43
2.2 Template matching and low-level approaches	46
2.2.1 Overview	46
2.2.2 Approaches	48
2.3 Aggregation-based methods	50
2.3.1 Overview	50
2.3.2 Approaches	50
2.4 Deep representation methods	52
2.4.1 Overview	52
2.4.2 Approaches	53
2.5 Method Comparison	56
2.6 Conclusion	58
3 NOVEL APPLICATIONS AND STUDIES USING CONTENT-BASED FEATURES	61
3.1 Hate Speech Detection of Social Media Content	61
3.1.1 Introduction and Overview	61
3.1.2 Problem Definition	62
3.1.3 Related Work	62
3.1.3.1 Text representations for Hate Speech	63
3.1.3.2 Classification approaches	64
3.1.4 Study and Proposed Method	64
3.1.4.1 Text representations	64

3.1.4.2	Classification Methods	66
3.1.5	Experiments and Results	67
3.1.5.1	Datasets and Experimental Setup	67
3.1.5.2	Results	68
3.1.5.3	Significance testing	70
3.1.5.4	Discussion	70
3.1.6	Conclusion and Future Work	71
3.2	Extractive Summarization of Web Documents	72
3.2.1	Introduction and Overview	72
3.2.2	Related work	73
3.2.2.1	Topic Modeling	73
3.2.2.2	Vector Space Models	74
3.2.2.3	Extractive Summarization	74
3.2.3	Proposed Method	75
3.2.3.1	Binary classification modelling	76
3.2.3.2	Topic-based Sentence Extraction	76
3.2.4	Experiments	78
3.2.4.1	Dataset and Preprocessing	78
3.2.4.2	Evaluation	79
3.2.4.3	TF-IDF Sentence Classification	79
3.2.4.4	Topic Modeling-based Classification of sentences	79
3.2.5	Results and Discussion	80
3.2.5.1	Classification Results	80
3.2.5.2	Rouge scores	82
3.2.6	Conclusions	82
3.3	Automatic Summarization of Video Game Reviews	85
3.3.1	Introduction	85
3.3.2	Related Work	86
3.3.2.1	Summarization Pipeline	87
3.3.2.2	Steam Review Summarization	88
3.3.3	Summarization Pipelines	89
3.3.3.1	CL pipeline	90
3.3.3.2	DL pipeline	92

3.3.4	Dataset	93
3.3.5	First User Study	96
3.3.5.1	Annotation Protocol	96
3.3.5.2	Participants	97
3.3.5.3	Results	97
3.3.6	Second User Study	99
3.3.6.1	Annotation Protocol	99
3.3.6.2	Participants	100
3.3.6.3	Results	100
3.3.7	Discussion	101
3.3.8	Conclusion	103
3.4	Clustering, Summarization and Classification of Web Documents and Social Media .	103
3.4.1	Introduction	103
3.4.2	Related Work	105
3.4.3	Approach	107
3.4.3.1	Change Detection Layer	109
3.4.3.2	Event Detection Layer	111
3.4.3.3	Semantic Layer	113
3.4.4	Experiments	116
3.4.5	Change detection	117
3.4.6	Event Detection	117
3.4.7	Conclusions	118
3.5	Scaling and Enrichment of Automatic Summarization	119
3.5.1	Introduction and Overview	119
3.5.2	Background	120
3.5.2.1	Related summarization systems and software	121
3.5.2.2	Sentence and information salience	122
3.5.2.3	Redundancy detection	126
3.5.3	NewSum: News Summarization in the real world	127
3.5.3.1	Real-world requirements	127
3.5.3.2	From n-gram graphs to Markov Clustering	129
3.5.3.3	N-gram graphs: the basics	129
3.5.3.4	Event detection as text clustering	131

3.5.3.5	Subtopic detection and representation	132
3.5.3.6	Measuring salience and avoiding redundancy	134
3.5.3.7	Incorporating entity information in n-gram graphs	134
3.5.3.8	Scaling n-gram graph-based analysis	135
3.5.3.9	NewSum: the architecture and the application	141
3.5.3.10	Evaluation of Summaries	143
3.5.4	Future Research Directions	146
3.5.5	Conclusion	147
3.6	Classifying Videos with Multimodal Deep Neural Networks	148
3.6.1	Introduction	148
3.6.1.1	Motivation	149
3.6.1.2	Problem definition	149
3.6.1.3	Structure	149
3.6.2	Related Work	150
3.6.2.1	Single Modality Video Classification	150
3.6.2.2	Audio-visual fusion	154
3.6.2.3	Contributions	155
3.6.3	Proposed method	156
3.6.3.1	Data preprocessing and frame encoding	156
3.6.3.2	Single-modality workflows	157
The FC workflow	157	
The LSTM workflow	159	
3.6.3.3	Multimodal workflows	160
Direct data fusion	160	
Sequence bias fusion	161	
Late video-level fusion	162	
3.6.4	Experimental results	164
3.6.4.1	Datasets and Experimental Setup	164
3.6.4.2	Single-modality experiments	166
Results	166	
Discussion	168	
3.6.4.3	Multimodal experiments	169
Results	169	

Discussion	170
3.6.5 Comparison to other systems	173
3.6.6 Conclusions	174
3.6.6.1 Summary	174
3.6.6.2 Findings	175
3.6.6.3 Future work	176
3.7 Conclusion and Findings	176
4 KNOWLEDGE-BASED REPRESENTATION ENRICHMENT: AN OVERVIEW	179
4.1 The need for enrichment	179
4.2 Knowledge Resources	180
4.3 Representation enrichment approaches	184
4.3.1 Input enrichment and modification methods	186
4.3.1.1 Overview	188
4.3.1.2 Approaches	188
4.3.2 Knowledge-based refinement methods	190
4.3.2.1 Overview	190
4.3.2.2 Approaches	190
4.3.3 Knowledge-aware end-to-end systems	192
4.3.3.1 Overview	192
4.3.3.2 Approaches	193
4.4 Conclusion	194
5 NOVEL APPLICATIONS AND STUDIES USING ENRICHED REPRESENTATIONS	197
5.1 Enriching Embeddings for Text Classification	197
5.1.1 Introduction and Overview	197
5.1.2 Related Work	199
5.1.3 Text preprocessing and embedding generation	203
5.1.4 Semantic enrichment	204
5.1.4.1 Semantic resource	204
5.1.4.2 Disambiguation	206
5.1.4.3 n -level hypernymy propagation	207
5.1.4.4 Fusion	209
5.1.5 Training	211

5.1.6	Workflow summary	211
5.1.7	Experimental evaluation	213
5.1.7.1	Datasets and experimental setup	213
5.1.7.2	Results	214
5.1.8	Discussion	227
5.1.8.1	Addressing the research questions	227
5.1.8.2	Comparison to the state-of-the-art	229
5.1.8.3	Execution runtime requirements	232
5.1.9	Conclusions	232
5.2	Enriching Embeddings for Automatic Text Summarization	233
5.2.1	Introduction and Overview	234
5.2.2	Related work	234
5.2.2.1	Text representations	234
5.2.2.2	Extractive summarization	235
5.2.2.3	Semantic enrichment	236
5.2.3	Proposed Method	236
5.2.3.1	Problem definition	236
5.2.3.2	Text representation	236
5.2.3.3	Semantic representation	237
5.2.4	Experiments	237
5.2.4.1	Datasets	237
5.2.4.2	Setup	238
5.2.4.3	Results and discussion	239
5.2.5	Conclusions	243
5.3	Conclusions and Findings	244
6	APPLICATION IN THE INDUSTRY	247
6.1	Problem Definition and Goals	247
6.2	Dataset	247
6.3	Proposed Method	248
6.3.1	Lexical Processing	248
6.3.2	Semantic Enrichment	249
6.3.3	Learning approach	250

6.4	Tuning and Monitoring	250
6.5	Implementation, Development and Deployment	250
6.6	Contributions	251
7	CONCLUSIONS AND FUTURE WORK	253
7.1	Summary and Contributions	253
7.2	Future Work	254
ABBREVIATIONS - ACRONYMS		257
APPENDICES		258
A	APPENDIX	259
A.1	20Newsgroups and Reuters dataset label names	259
REFERENCES		319

LIST OF FIGURES

2.1 Low-level representations	44
2.2 Aggregation-based representations	44
2.3 Deep representations	45
3.1 The pipeline for the TF-IDF-based extractive summarization.	75
3.2 Pipeline for topic-modeling based extractive summarization.	76
3.3 Proposed summarization pipeline variants	89
3.4 User interface for online evaluation of summaries produced by CL AsDe and CL Full methods.	96
3.5 Change Detection example	104
3.6 Geosensor Event Detection example	107
3.7 The system architecture of GeoSensor.	108
3.8 The workflow implemented by Change Detector.	110
3.9 The Spark-based implementation of Event Detector.	112
3.10 Entity extraction example	114
3.11 User criteria for triggering (a) Change Detection, and (b) Event Detection. .	115
3.12 Parallelization approaches execution times	116
3.13 Event Detector Spark implementation performance	118
3.14 From string to n-gram graph	133
3.15 Graph operations performance versus SPARK partition count	136
3.16 Average time elapsed for merging 2 topics (left), merging 4 topics (middle) and feature extraction.	137
3.17 ARGOT vs the multithreaded JINSECT implementation	137
3.18 The BDE event detection process	137
3.19 Event Detection workflow Similarity Mapping procedure	139
3.20 Experimental results on the similarity mapping performance	140
3.21 NewSum snapshots	143
3.22 Preliminary “Open Beta” summary grades	145
3.23 User-aware evaluation summary grades	145

3.24 Visual content preprocessing	157
3.25 Audio content preprocessing	158
3.26 Frame encoding output	159
3.27 Early and late frame fusion	159
3.28 The LSTM workflow	160
3.29 RNN image description example	161
3.30 The audiovisual input-bias fusion	162
3.31 The input-bias multimodal fusion method	162
3.32 The state-bias multimodal fusion method	163
3.33 Multimodal late fusion	163
4.1 Input modification	185
4.2 Knowledge-based refinement	185
4.3 Knowledge-aware end-to-end systems	186
5.1 Semantic augmentation approach overview	199
5.2 The <i>basic</i> disambiguation strategy	204
5.3 The <i>basic</i> disambiguation strategy	205
5.4 Vector construction phase of the <i>context-embedding</i> stragegy	208
5.5 Disambiguation phase of the <i>context-embedding</i> stragegy	208
5.6 Example of the spreading activation process	210
5.7 Semantic augmentation process example	212
5.8 Confusion matrix 20Newsgroups	219
5.9 Confusion matrix Reuters	224

LIST OF TABLES

2.1 Non-enriched representation methods	47
2.2 Comparison between representation approaches	57
3.1 Performance of feature approaches	69
3.2 Classifier ANOVA results	70
3.3 Tukey's HSD test results	71
3.4 Multiling 2015 MSS dataset	78
3.5 TF-IDF sentence classification results.	79
3.6 Micro/macro F1 performance comparison	80
3.7 Topic modeling results in micro and macro F1 score.	81
3.8 TF-IDF Rouge Scores	83
3.9 Topic modeling Rouge Scores	84
3.10 Aspects and keywords used for the identification of dominant aspects in review clusters.	91
3.11 Summary generation examples	94
3.12 Games selected from the dataset	95
3.13 First user study	98
3.14 Analysis of variance	98
3.15 The datasets used in the experimental evaluation.	165
3.16 Single-modality collective results for all datasets.	167
3.17 Multimodal collective results for all datasets.	170
3.18 Multimodal fusion method average ranks.	171
3.19 Multimodal fusion workflow comparison	171
4.1 Knowledge resources	181
4.2 Enriched representation methods	187
5.1 Dataset characteristics for 20Newsgroups and Reuters	214
5.2 Main experimental results on 20Newsgroups	217
5.3 20-Newsgroups misclassification examples	218

5.4	20-Newsgroups t-test results	220
5.5	Performance on 20Newsgroups with a concept-wise frequency threshold .	220
5.6	Performance on 20Newsgroups with a dataset-wise frequency threshold .	221
5.7	Reuters misclassification examples	223
5.8	Main experimental results on Reuters	224
5.9	[Reuters t-test results	225
5.10	Performance on Reuters with a concept-wise frequency threshold	226
5.11	Performance on Reuters with a dataset-wise frequency threshold	226
5.12	Comparison to the state of the art	230
5.13	Comparisons with additional datasets	231
5.14	Multiling 2015 MSS dataset	238
5.15	MultiLing2015 word2vec CBOW results	241
5.16	MultiLing2015 FastText results	242
6.1	HSD Dataset	248
A.1	Label indexes to name mapping.	260

1. INTRODUCTION

1.1 Machine Learning in the Modern Digital Landscape

A vast amount of different types of data are prevalent in our digital media ecosystem. Large quantities of content, including text, images, audio and video, are available and circulated on the Internet, ranging from journalism websites, blogs and academia-related content, to fiction literature portals and social media. The efficient management, browsing and consumption of this content depends on accurate discovery and search operations, applied on massive data collections. To this end, the development of robust machine learning methods have been crucial in the era of big data. Such methods include classification systems for data tagging and categorization [9, 608], that automatically assign labels to new instances, facilitating the efficient organization and categorization of large volumes of data with little to no human involvement. In addition, clustering techniques [276] group together data without the need of annotations, considering features, attributes and characteristics of data instances instead to produce cohesive clusters. Further, summarization solutions [690] focus on text, aiming to automatically extract informative summaries to aid the fast and efficient navigation of documents.

Such machine learning systems are applied to a wide array of commercial, industrial and artistic applications, from phishing and spam detection [661, 602], medical imaging, style transfer and optical character recognition [521, 288, 581], speaker diarization and genre classification [21, 600], news and social media summarization [39, 113], recommender systems [219] and image processing [116]. The ubiquitous adoption of such models has enabled the automation of such tasks, avoiding the comparatively prohibitive cost of manual human effort on such a large scale. It is thus of critical importance that these systems operate in an efficient and robust manner; an avenue towards achieving such performance enhancements is the topic of this thesis, which will be investigated in the sections and chapters that follow.

1.2 Structure of a Machine Learning System

A typical machine learning system is composed of the following components:

1. Inputs: A dataset D , composed of input instances $d_i, i = [1, 2, \dots, N]$, corresponding to real world objects processed by our system (e.g. documents, images, audio or genetic sequences). For supervised tasks, instances d_i are accompanied by ground truth elements $L_i, i = [1, 2, \dots, N]$.

In finite ground truth sets (e.g. classification), $L_i \in L$ with L being the set of all available ground truth annotations in the dataset. An annotation $l \in L$ is a semantic tag related to the content of d_i , either directly (e.g. topic or sentiment classification) or indirectly (e.g. related to content generation and details of the task at hand,

such as in authorship attribution, style classification tasks and extractive sentence-based summarization). We will refer to finite ground truth sets as labelsets that contain individual labels, for the remainder of the study. Supervised problems can be characterized by the number of possible labels $|L|$ an instance can be matched with – common paradigms are binary classification tasks (two available labels, often corresponding to a “yes” or “no” answer, e.g. hate speech detection, medical image disease prediction, speech pathology detection, extractive summarization) and multiclass classification (more than two candidate labels, e.g. as in text topic classification, visual object recognition, and music genre classification). The maximum number of label annotations per instance $M = \max_{i=1\dots N} |L_i|$ renders the task as a single-label ($M = 1$, i.e. only one label allowed per instance – e.g. sentiment analysis, handwritten digit recognition and speaker diarization) or a multi-label classification problem ($M > 1$, i.e. multiple possible labels per instance – e.g. document topic classification, visual object recognition and music genre classification).

For tasks with non-finite ground truth sets, annotations can be continuous values within a specified range (e.g. probability scores or normalized weights), or vary without any restrictions (e.g. values in \mathbb{R} . Examples may include extractive summarization tasks with a ranking / importance score per sentence.

2. Preprocessing: a set of preliminary preprocessing operations applied to the data [193, 641, 75, 89, 108], such as:

- Data Augmentation: when dataset-related limitations impact performance (e.g. sample scarcity, label imbalance, etc.) operations that modify the input collection may be employed, such as data augmentation and under/oversampling [635, 238, 572].
- Data cleaning [112] – e.g. handling undesirable input patterns (e.g. non-alphanumeric / whitespace in text, image / audio denoising, frequency filtering, etc. [58, 84])
- Filtering – e.g. stopword handling, stemming, lemmatization, POS extraction in text [576, 289, 519], normalization / equalization in images / audio [491, 198, 58, 32, 230] as well as modality-shifting operations [316].
- Segmentation – e.g. word / sentence splitting and tokenization in text [641], region / color segmentation in images [58], temporal / spectral segmentation and source separation in audio [700, 97, 431, 617]

3. Representation: A representation mapping [418] converts preprocessed inputs into a suitable format, with respect to computational costs of subsequent processing and learning efficiency [52]. This is often realized through mappings to a vector space [545] although different approaches have been explored [589, 291, 208].

Rather than treating an input instance as a whole, it is often useful to segment it, breaking it into multiple subproblems that are handled separately. This enables tackling the representation extraction problem in different granularity levels and points

of interest and can be beneficial in terms of computational complexity, since the input size can be greatly reduced. Additionally, it can positively affect system performance when the segmentation is content-based (e.g. word and sentence splitting, image color / texture-based segmentation and audio source separation), rather than naive or based on simple heuristics (e.g. spatial / pyramidal image decomposition / temporal subdivision of audio, etc.). Such divide-and-conquer approaches [561] are finalized by aggregating the generated subproblems towards arriving at a single, final representation for the input instance, utilized by subsequent stages of the learning pipeline.

4. Learning: the next step involves training a machine learning system to the represented data in order to solve the task at hand. Training quality is measured by appropriate evaluation measures for the task [588, 517, 408]. After training, the model can be evaluated on data not encountered during training, in order to assess its generalization ability [446].

Given this framework, in the next section we briefly identify limitations and potential for improvement for machine learning tasks, by utilizing enrichment approaches.

1.3 Enriching Data Representations

As examined in the previous section, an important machine learning component is building data representations, i.e. mapping real-world objects processed by the system (e.g. emails, photographs, audio recordings) into a feature collection that can be processed by the computer and fed to a machine learning workflow [598]. Since this mapping is often the only source of information for subsequent parts of the workflow, producing high-quality representations is a crucial component for efficient learning.

This often requires the construction of representations that go beyond low-level local pattern-matching (e.g. token/ngram frequencies in text, local template matching in audiovisual content), but encapsulate complex, conceptual information [48, 52]. A lot of research effort has pursued building representations via handcrafted feature engineering and transformation, as well as automated methods for content-based representation learning [716, 47]. However, engineered approaches tend to rely on empirical expert knowledge specific to the modality, data, domain and/or task at hand and is often based on rigid heuristics (e.g. text token bag hyperparameters, visual descriptor templates or audio signal/temporal/segmentation-based measures).

Recently, rapid advances in representation learning via deep learning methods and large neural network systems [467, 10] strive to build conceptual knowledge from the bottom up. To achieve this however, these approaches operate on vast amounts of data, require considerable computational resources and energy [599] and heavily rely on the distributional hypothesis [235] to arrive at sets of features that hopefully encapsulate useful semantics.

In order to address the limitations of the aforementioned approaches, we explore a differ-

ent approach for arriving at representations that encapsulate useful, high-level information: in this study we propose enriching content-based representations by directly exploiting existing resources of structured, encoded human knowledge. Along with the rapid growth of available data, a systematic collection, structured formatting and storage of knowledge has accompanied the growth of artificial intelligence research and the development of commercial AI-powered solutions. As a result, a wealth of curated, high quality information is available and applicable in machine learning systems and machine learning tasks, ranging from fine-grained linguistic and audiovisual information [190] to high-level conceptual ontologies [441, 143, 516, 194]. However, the utilization of such resources for broad-range solutions has been lacking.

The main focus of this study is using resources of high-quality information and knowledge (e.g. conceptual, semantic, relational) and investigating methods of injecting it into data representation methods for improving the performance of different machine learning tasks for data of different modalities. In contrast with contend-based paradigms, this enrichment aims to supply representation construction methods with ready-to-use high-level information, the utilization of which will result in a representation that is better suited for the machine learning task of interest.

The study is formulated through a brief survey of representation methods used for different machine learning tasks, modalities, representation approaches, knowledge resources and enrichment avenues. It involves overviews of existing approaches in the literature for content-based representations, knowledge resources and enrichment methods, as well as novel applications and proposed representation enrichment systems.

1.4 Contributions

This work explores avenues of exploiting sources of structured human knowledge (currently ignored or under-utilized in machine learning applications) towards improving performance. Our specific contributions include:

- An overview of the literature for representation extraction approaches, spanning different paradigms and modalities of data, with a focus on the classification setting.
- A set of novel applications of different representation methods for different tasks (e.g. classification, clustering and summarization) and data modalities (e.g. text, image and audio).
- A comparative presentation of resources of encoded human knowledge, that can be used for representation enrichment in machine learning problems.
- An overview of approaches in the literature for representation enrichment, utilizing resources of human knowledge, exploring different data modalities and representation paradigms, but focused on classification.
- A presentation of proposed methods for representation enrichment for classification and automatic summarization tasks for text, with detailed, large-scale experimental

evaluations and analyses.

- Suggested future directions in representation enrichment, in light of the totality of the presented work.

1.5 Thesis structure

This study is structured as follows.

- In chapter 2, we present a review of different approaches for representation methods for machine learning tasks that are entirely content-based, i.e., they *do not* utilize sources of existing / extrinsic knowledge. We organize the related work for these approaches in three categories corresponding to the representation construction approach, with each category arriving at features with different richness of information.
- Next, in chapter 3 we present novel applications, investigations and/or extensions of various representation approaches, utilized on different machine learning tasks and data modalities.
- In 4, we move on to consider knowledge resources and use the information contained within to enrich data representations. We provide an overview of enrichment approaches in the literature, organized in insightful categories pertaining to the enrichment scheme.
- Chapter 5 focuses on proposed enrichment schemes of selected machine learning tasks, where we propose and evaluate novel methods for augmenting representations with human knowledge.
- Chapter 6 showcases a realization of the totality of research findings and insight produced in this study, with the implementation of a novel a Hate Speech Detection pipeline designed for a real-world industrial setting.
- We conclude the study in chapter 7, summarizing the covered material and the findings discovered. Additionally, we suggest directions for future work in machine learning representation enrichment.

2. CONTENT-BASED REPRESENTATION APPROACHES: AN OVERVIEW

In this chapter, we perform an investigation over different data representation approaches that exist in the literature, that do not explicitly consider external / extrinsic information – i.e., they produce the representation via a procedure that is entirely content-based and knowledge-agnostic. Regarding content-based representations, relevant surveys and literature reviews focus on specific data modalities, e.g. covering text data [565, 443, 655], image data [631] and audio data [185], as well focusing on specific machine learning tasks [397, 718]. Here, we provide a cross-modal investigation focused on representations for the classification task, providing a comparative overview of different techniques and paradigms.

2.1 Categorizing content-based approaches

We structure the discussion by organizing the approaches into 3 groups of increasing level of abstraction, from low to high of information content and resulting representation generality, namely:

- **Local template-matching and low-level representations:** in this category we cover approaches that rely on matching predefined templates on the input data. The collection of responses of the template matching constitutes the representation output and a corresponding vector embedding space. Such approaches are covered in section 2.2, and include, e.g., bags of words and features, audio and visual descriptor templates, statistical/signal measures of data streams, etc. [565, 295, 77]. A visualization of the category is available in Figure 2.1.
- **Aggregation-based methods:** here we include approaches that group, transform and/or combine low-level representations from the previous category. Apart from combining / transforming the input into representations that contain higher-level information, such operations are applied for computational efficiency, redundancy filtering and dimensionality reduction, and may include clustering, topic modelling and decomposition methods [633, 530, 433]. They are covered in section 2.3 and a visualization is presented at Figure 2.2.
- **Deep representations:** this group covers approaches that heavily rely on a hierarchy of modular, non-linear components for representation learning. We focus on neural network models and deep learning that can produce very high-level information, i.e. rich features that correlate to high-level abstract / conceptual / context-aware information. These methods, including convolutional, recurrent and transformer neural networks [427, 227, 614] are presented in section 2.4 and can be visualized in Figure 2.3.

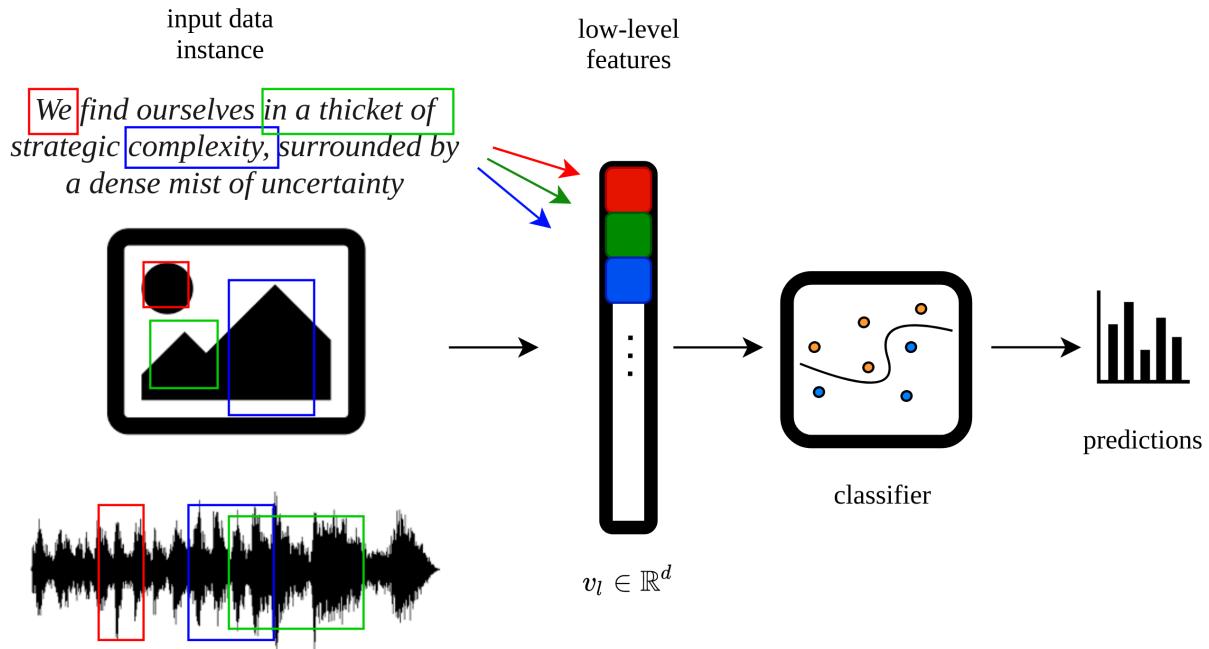


Figure 2.1: Local template matching for production of low-level representations. Feature extraction templates are applied to extract responses from local regions of the input data (e.g. text, image or audio), subsequently composed into d -dimensional feature vectors, subsequently fed to a classification machine.

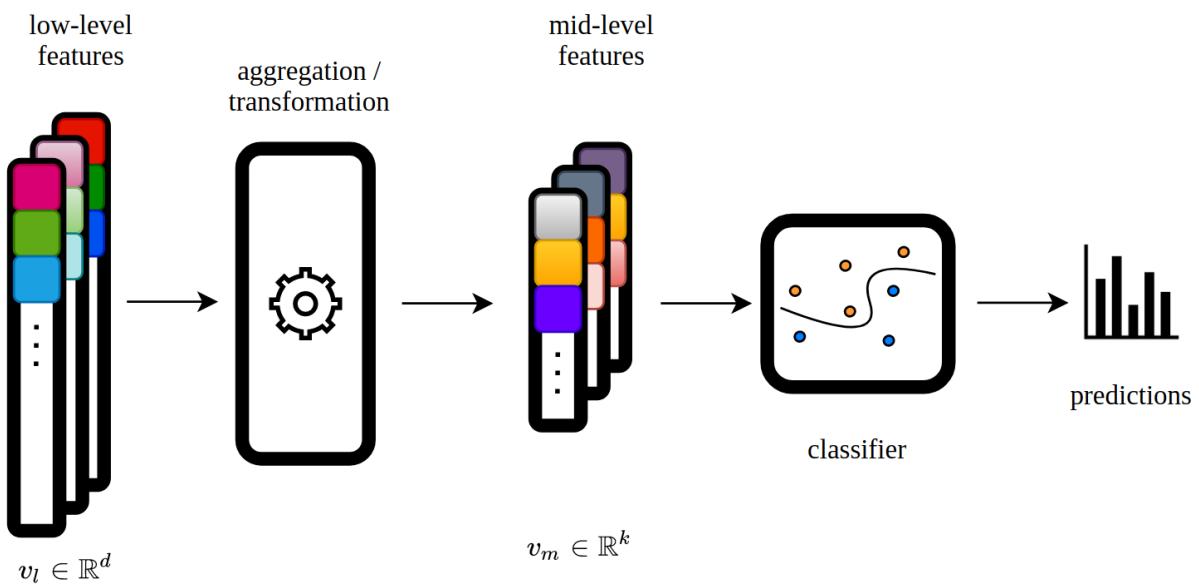


Figure 2.2: Aggregation-based representations combine, fuse, transform and/or extract low-level features via static engineered procedures that implement a mapping $T : \mathbb{R}^d \rightarrow \mathbb{R}^k$, (often with $k < d$). The aggregated features are subsequently used for prediction via a classifier.

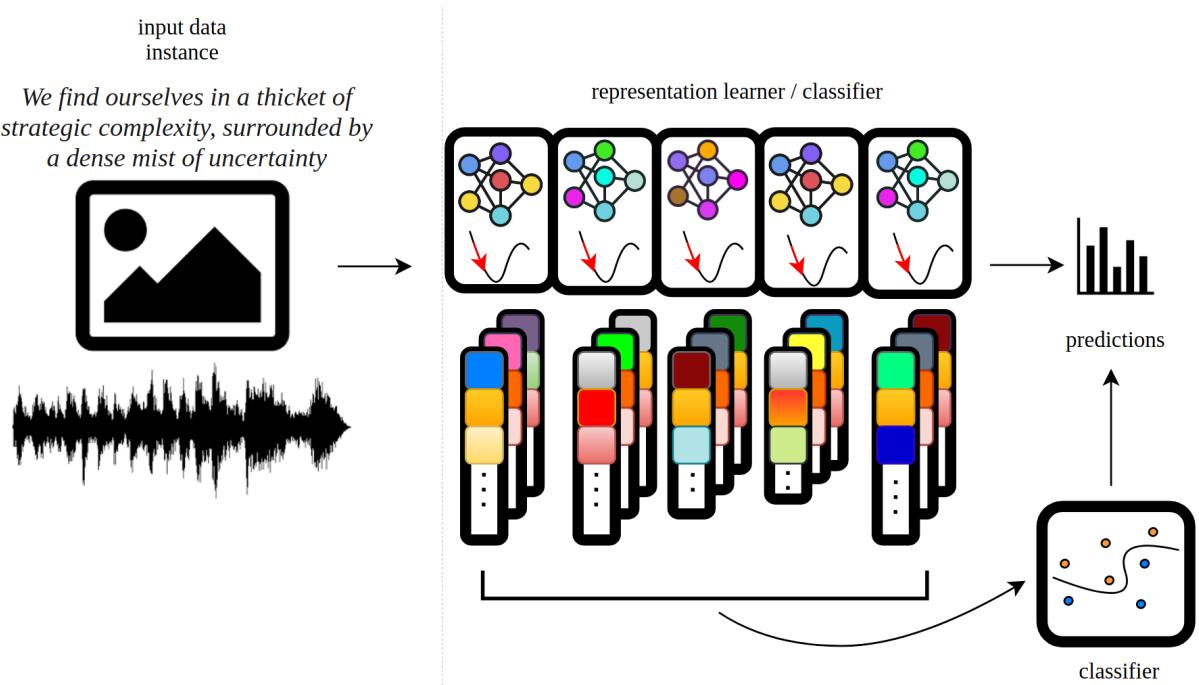


Figure 2.3: Deep representations rely on hierarchical non-linear architectures that perform end-to-end representation learning, often composed of neural network-based architectures trained with a loss function to be minimized (as depicted via the shapes in each learning chain block / layer). Layers in the architecture produce feature sets at multiple levels, information richness and abstractness. Deep models can be jointly trained for both representation learning and classification (top pipeline) or used as feature extractors in conjunction with other classifiers (bottom pipeline).

For all three representation categories, we consider works dealing with text, images or audio data, modalities that present semantic gaps of different severity: for text, high-level semantic concepts have to be retrieved from words, sentences or paragraphs, i.e. tokens that result from semantic segmentation performed by humans, readily available and delineated by virtue of grammatical and syntactic rules in language. In contrast, content-based approaches for images and audio can only extract primitive features from binary digital data (i.e. visual / audio signal values) with little to no semantic content or direct linkage to high-level information [569].

The sections that follow cover each representation category, presenting related work for different modalities and tasks that adopt such approaches, with a primary focus on classification. In section 2.2, we cover low-level representations that rely on template matching, followed by an examination of aggregation-based methods in section 2.3. Finally, deep representation approaches are examined in section 2.4. An overview of the material covered is presented in table 2.1, where for each listed study we report the modality (text, images or audio), the category (as listed above), the representation approaches, the labelling scheme, the utilized learners and adopted evaluation metric(s).

2.2 Template matching and low-level approaches

In this section we cover representations that rely on matching templates on the input data [52], producing low-level information as a response.

2.2.1 Overview

Given a representation template, we can discriminate between “local” (i.e. the template is applied on a subsection of the input) and global (i.e., the template covers the entire input instance) applications of the extraction process [707, 435]. Local templates may regard the input as a collection of key-value pairs: keys locate distinct attributes (e.g. an individual word / ngram in text, a point of interest in images and audio, etc.) while values correspond to a magnitude of match or *weight* of the template in that location. Terms may be easily delineable in the source data (e.g. individual words in text, detected keypoints in images, specific temporal slice / peak in audio) or lack high-level semantics and an intuitive explanation – the latter can affect the interpretability of the representation and, down the line, of the entire learning pipeline [155, 134]. Given collections of term-weight pairs, feature selection approaches may be applied for filtering purposes [684, 96, 228]. Global features usually provide coarse information on a narrow view of the input – e.g. color distribution information in images, sentiment / grammaticality scores in text, SNR values for audio, etc. [473].

citation	mod.	category	representation	labelling	classifiers	metrics
[684]	TXT	LOW	BoW, Feat-Sel	MC, SL/ML	SVM	AP
[290]	TXT	LOW	BoW	ML	Rule	PR, RE
[563]	TXT	LOW	BoW	MC-SL	SVM	ACC
[365]	TXT	AGGR	BoW, PCA, CLUST	MC-SL	NB, k-NN, DT, ENSEMBLE, SUBSPACE	ACC
[620]	TXT	AGGR	TFIDF	MC-SL	k-NN	ACC
[712]	TXT	AGGR	TFIDF, LSA, CLUST	MC-SL	SVM	ACC
[381]	TXT	AGGR	BoW, LSA	MC-SL	SVM	F1
[704]	TXT	AGGR	BoW, LSA	MC-SL	cosine	ACC
[677]	TXT	AGGR	BoW, PLSA	MC-SL	SVM, NB, KDE	ACC
[444]	TXT	AGGR	BoW, LSA	MC-SL	kSVM, NB, k-NN	ACC
[567]	TXT	AGGR	TFIDF, PCA	MC-SL	MLP, cosine	P, R, F1
[208]	TXT	AGGR	NGG	MC-SL	similarity	P
[682]	TXT	DEEP	SKIPGRAM	BIN	NEURAL	ACC
[103]	TXT	DEEP	EMBEDDINGS	MC, SL	SVM, LR	ACC
[382]	TXT	DEEP	SKIPGRAM, LDA	MC, SL	LINEAR	PR, RE, F1, ACC
[118]	TXT	DEEP	CONV	BIN / MC, SL	NEURAL	WER
[337]	TXT	DEEP	CONV	MC, SL	NEURAL	F1
[404]	TXT	DEEP	BoW, NLM	BIN / MC, SL	SVM	ACC
[347]	TXT	DEEP	SKIPGRAM	MC, SL	MLP	ACC
[604]	TXT	DEEP	TRANSFORMER	BIN / MC, SL	NEURAL	ACC
[498]	IMG	LOW	LBP, Gabor	Binary	k-NN	ACC
[356]	IMG	LOW	Color, Gabor	MC-SL	k-NN	ACC
[656]	IMG	LOW	SIFT, Shape	MC-SL	k-NN	ACC
[60]	IMG	LOW	SIFT, LBP	MC-SL	KELM	ACC
[683]	IMG	AGGR	SIFT, Gabor, KMeans	MC-SL	SVM, MAP	ACC
[191]	IMG	AGGR	SIFT, PCA, KMeans	MC-SL	kSVM	ACC
[71]	IMG	AGGR	SIFT, HOG, SPM	MC-SL	RF	ACC
[680]	IMG	AGGR	SIFT, SPM, sparse coding	MC-SL	SVM	ACC
[392]	IMG	AGGR	SIFT, Kmeans, VLAD	MC-SL	SVM	ACC
[494]	IMG	AGGR	SIFT, color, GMM, Fisher Kernels, PCA	MC-SL	SVM, SLR	AUC, F1
[721]	IMG	AGGR	SIFT, LBP, SPM, KMeans	MC-SL	KCR	ACC
[324]	IMG	DEEP	CONV, POOL, NORM	MC, SL	NEURAL	ACC
[580]	IMG	DEEP	CONV, POOL, NORM	MC, SL	NEURAL	ACC
[610]	IMG	DEEP	INCEPTION	MC, SL	NEURAL	ACC
[239]	IMG	DEEP	INCEPTION, RESIDUAL	MC, SL	NEURAL	ACC
[263]	IMG	DEEP	CONV, RESIDUAL	MC, SL	NEURAL	ACC
[663]	AU	LOW	signal, psych., spectral	MC-SL	distance	ACC
[344]	AU	LOW	signal, musical, psych.	MC-SL	kSVM	ACC
[399]	AU	LOW	signal, spectral	MC-SL	k-NN, rule-based	ACC
[400]	AU	LOW	MFCC, signal, spectral	MC-SL	SVM, k-NN, GMM	ACC
[229]	AU	LOW	MFCC, signal, spectral	MC-SL	SVM	ACC
[370]	AU	LOW	signal, spectral	MC-SL	SVM	P, R, F1, ACC
[426]	AU	AGGR	signal, spectral, psych.	MC-SL	Gaussian QDA	ACC
[310]	AU	AGGR	MFCC, VQ, LDA	MC-SL	kSVM	F1
[351]	AU	AGGR	MFCC, GMM, PLSA	MC-SL	kSVM	AP
[243]	AU	AGGR	signal, spectral, events	Binary	SVM, k-NN, LogReg, NB, DT	ACC
[35]	AU	AGGR	MFCC, signal, spectral, musical, PCA, MRMR	MC-SL	SVM	ACC
[350]	AU	DEEP	CONV, DBN	MC, SL	SVM, GDA, k-NN, ENSEMBLE	ACC
[705]	AU	DEEP	CONV, POOL	MC, SL	LINEAR	ACC
[266]	AU	DEEP	CONV, POOL, ENCDEC	MC, SL	SVM	ACC
[107]	AU	DEEP	CONV, POOL	MC, SL	NEURAL	ACC
[242]	AU	DEEP	CONV, INCEPTION, RESIDUAL	ML	NEURAL	mAP, AUC

Table 2.1: Outline of knowledge-agnostic representation methods. LOW, AGGR and DEEP refer to the categories outlined in section 2.1. MC, SL and ML labelling refers to multiclass, single-label and multi-label configurations respectively. Evaluation measures ACC, PR, RE, F1, AUC and AP refer to accuracy, precision, recall, F1-measure, area under curve and average precision, respectively. Entries in the representation / classifier columns refer to acronyms described in the text, or descriptive categories of algorithms and approaches. Classifiers NEURAL refer to applying the neural model itself with an appropriate output layer (e.g. a d -dimensional fully-connected linear layer followed by a softmax function, for the prediction of d output classes).

2.2.2 Approaches

A popular representation approach utilized in classification is using vector space embeddings, where a Vector Space model (VSM) [545] projects data into a vector $v \in \mathbb{R}^d$, which can be manipulated with distance measures and Linear algebra constructs [40, 95] in order to process and compare instances. The Bag of Words (BoW) model [544, 565] and Bags of Features (BoF) approaches in general, are popular low-level VSMs that produce count-based weights for points / regions of interest in the input.

BoF is a popular baseline for text, where semantically salient terms are easily identifiable and delineated by syntax and grammar. Common weighting schemes include boolean, term and document frequency (BF, TF, DF), which denote presence, instance-level and collection-level counts of a term in the text. BoF methods have been widely used for document classification. In [365] the authors use bags of words and bigrams for news article categorization, producing a term-document count matrix. Similarly, in [290] a BoW model is used to classify news and biomedical data with support vector machines (SVM) [636, 240], using stopword removal and stemming preprocessing. In [620], the authors use term frequency - inverse document frequency (TF-IDF) weighting [544], which normalized counts by their DF weight. They proceed to classify articles to four categories using k-NN. The work in [684] uses BoW followed by different feature selection methods (e.g. DF thresholding, information gain (IG), χ^2 -test and term strength). Instances are classified with k-NN and Linear Least Squares Fit models. The work in [563] explores phrases and sentences as the terms of interest, implementing classification with IG-based rules. To deal with the exponential number of the candidate sets, they utilize a selection process via noun-phrase and key-phrase extraction.

In the visual domain, since images lack easily identifiable semantic boundaries, and pixel-level approaches are intractable for the majority of real-world classification problems, local methods attempt to break down content into useful subdivisions (similar to words / phrases in text). This process consists of a) selection of keypoints, i.e. salient regions and/or positions of interest, and b) visual content description, in the proximity of these locations. Keypoint detectors highlight corners, blobs, edges, or are sampled from a grid [41, 234, 395, 570, 436, 628], while descriptor templates map keypoint-adjacent regions to numeric vectors that summarize their content [328, 295, 395, 41], capturing variations in intensity, direction, flow, etc., and retaining desirable properties (e.g. location/rotation/scale invariance). On the other hand, global features produce coarse-grain descriptions of the entire image, e.g., color histograms, moments, layout, etc.; see [506] for an overview). Extensions of visual descriptor methods include applications to separate color channels [415], combining local and global features [375, 372], adopting segmentation / multi-resolution steps [223], etc. – see [629, 384] for a survey.

Regarding specific approaches of visual representations for classification, in [683], the authors utilize SIFT [395], a popular descriptor invariant to operations such as illumination and viewpoint shifts. Additionally, the authors use gabor features [410], extracting texture-related information at different scales and orientations. The representations are fed to SVM and MAP classifiers to predict land cover classes from satellite images. Likewise,

SIFT features are utilized in [191] for descriptions of patches in a uniform grid (instead of keypoint detection) and are classified through RBF kSVMs. Additionally, SIFT is utilized in [656] for leaf classification, in conjunction with shape context features [547] and k-NN matching. In [498], color histograms on the RGB and Ohta colorspaces are used as low-level features, along with histograms of Local Binary Pattern (LBP) [471] responses applied both on separate color channels and the entire RGB volume of the input image. The features are used towards texture classification with k-NN. In [356], gabor texture features [428] are used for natural image classification with k-NN, where color channels are individually exploited for feature extraction. The work in [60] uses dense eSIFT [94] in a spatial pyramid matching (SPM) setup [346], combined via linear combination with global feature responses, i.e. Local Binary Pattern [471] over multiple scales, focal points, orientations, etc. KELM [264] is used on the fused representations to classify aerial scene images.

For the audio modality, descriptors process content in the temporal or frequency domain and include signal-based and statistical measures, psychoacoustic features, musical (e.g. tone, timbre, rhythm, pitch, etc.) descriptors, decomposition methods, and others [630, 490, 596, 528, 344, 77, 583].

Regarding specific approaches, in [663], loudness, pitch, brightness, bandwidth and harmonicity features are used to compose instance vectors from audio clips, classified with respect to Euclidean distance to the mean of manually selected class representatives. In [399], features such as high zero-crossing rate, low short-time energy ratio, spectrum flux, band periodicity, noise frame ratio and linear spectral pairs are used to discriminate speech and non-speech content, followed by a rule-based final categorization to four audio classes. In the work of [426], low-level signal properties, MFCC [583], psycho-acoustic features (e.g. audio sharpness, roughness, etc.) and temporal envelope processing features which mimic human auditory processing, are generated through a series of predefined filter banks. Gaussian quadratic discriminant analysis classification is used to categorize music into genres. Similar features (i.e. statistical measures over signal-based features) are used in [229] and [400] – in the former, SVMs are used to classify sounds from a diverse set of 16 categories. In the latter, SVMs, k-NN and Gaussian Mixture model (GMM) [523] approaches are used to categorize short audio clips to silence, music, background sound, pure and non-pure (e.g. noisy) speech. In [370], statistical, spectral, signal-level and psychoacoustic features are utilized for genre classification via SVMs. The work in [651] exploits MPEG-7 [309] descriptors and spectrum statistics for environmental sound classification. The labels include 12 types of home environment sounds (e.g. speech, animal sounds, various effects). SVM and k-NN classifiers are used, fusing their outputs to a single final prediction. In [344], signal-based, musical (tonal, timbral) and psychoacoustic features are used for multiclass emotion recognition in audio. The authors produce correlation measures of the target classes with feature categories via SVM and Linear Regression, as well as evaluating the features in emotion classification with an RBF-kernel SVM. The authors in [243] tackle folk song classification, proposing a probabilistic event modelling framework that models audio sequences akin to a language model [82]. To handle sparsity in the event classes and facilitate, e.g., pitch and duration mod-

elling, they cluster the latter into characteristic abstract constructs, i.e. “viewpoints” [119]. They compare this approach to four different collections of global audio features, ranging from pitch and duration statistics to harmonicity and musical scale templates. Multiple classifiers are utilized, including SVM, k-NN, Naive Bayes (NB) [525], Logistic Regression and Decision Trees (DT) [358].

In summary, this section showcased approaches that utilize low-level information for classification of text, image and audio data; we move on to examine approaches that transform, manipulate and aggregate this information towards improving performance and tractability of the classification problem.

2.3 Aggregation-based methods

In this section, we explore approaches that rely on aggregating, combining and/or transforming lower-level representations to arrive at higher-level features, with respect to the abstractness and richness of the information encapsulated.

2.3.1 Overview

Aggregation methods produce mid-level representations from low-level and/or primitive inputs, using engineered and/or learned complex functions rather than directly exploiting token-based statistics. This additional step in the representation process comes with an increase in the computational cost of the approach, but can offer multiple benefits. Contrary to low-level features, aggregation methods usually build distributed representations [248, 534]: i.e., resulting semantics are spread or “distributed” over multiple dimensions in the embedding space [52], arriving at compact and robust representations but sacrificing explainability. Dimensionality reduction is often facilitated by these approaches, while maintaining or improving the expressive power of the output features, mitigating the problem of the curse of dimensionality [46]. In light of these remarks, we present related work utilizing this representation paradigm, with covered studies being summarized in table 2.1.

2.3.2 Approaches

For text data, local low-level methods deal with distinct words, characters, and/or respective ngrams; as a result, the generated vocabulary can quickly scale to very large sizes, reducing system performance and accuracy [50]. Thus, aggregation methods for text will often aim to shift the representation from the space of the vocabulary to one that is denser, more compact and simultaneously preserves the majority of the information content.

For example, early works [365] post-process term count features with Principal Component Analysis (PCA) [292] and apply hierarchical clustering to tackle news categorization. Multiple learners are evaluated, such as NB, k-NN, DT and a feature subspace approach.

In [712], TF-IDF features, a multi-word term vector approach [298] and coefficients from Latent Semantic Analysis (LSA) [139] are compared for categorization of news articles and academic journals with linear SVMs. In [381], the authors compute LSA vectors on a local document-term space deemed most relevant to each class, as determined by a preliminary classification phase. The final categorization is performed with SVMs to classify web documents and webpages to a variety of topics. In [704], unlabeled data are used to alleviate data scarcity and expand the term-document matrix, prior to LSA decomposition. Instance vectors are classified via cosine distance to multiple classes and domains. LSA decomposition is also applied in [444], on academic document titles. The classification component uses an RBF-kernel SVM, k-NN and NB to classify the generated LSA coefficient vectors to thematic categories (e.g. engineering, mathematics, history, etc.). In [567], the authors apply PCA and a manually curated, entropy-based weighting, to TF-IDF features, used to classify web pages via a shallow neural network or the cosine distance. A topic-based approach is employed in [677], with Probabilistic LSA (PLSA) [253, 254] topics being used as a cross-domain “bridge”, extending supervised models with information from unlabelled data. SVM, NB and Kernel Density Estimation [696] are used for predicting news and forum post categories. In [208], n-gram graphs (NGGs) [201] are used to create aggregate representative graphs for each class. These are then compared with document NGGs for categorization via different similarity measures of various web documents.

For images, the popular approach of local descriptors generally yields non-scalar information, resulting in large volumes of data. Thus, many approaches pool together multiple descriptor responses to reduce the load fed to subsequent classification components, filter out redundant / superfluous information and alleviate noise.

For instance, in [683], an aggregate vector is built by vector quantization [197] and clustering. Local SIFT responses are clustered via K-means [276], each subsequently assigned to the closest cluster, building a visual bag-of-features vector. In [191], SIFT responses are processed by PCA [292], and subsequently clustered into a vocabulary of visual words via vector quantization with KMeans. This generates a global bag vector for the entire image that is classified with an RBF kSVM. The work in [71] uses SPM to create BoF multiscale vectors for image subregions, using grid-sampled SIFT variants and HOG [129], in a local and global configuration. Classification is implemented via random forests and ferns [476]. In [680], a SIFT-based codebook is built via sparse coding, which produces sparse rather than dense quantized descriptor vectors. This is followed by multiscale max pooling and linear SVM classifiers for scene and event categorization. The work in [392] utilizes Vectors of Locally Aggregated Descriptors (VLAD) [282, 141], where assignments to KMeans-clustered visual words are not binary, but implemented via instance-cluster residuals. Linear SVMs are used to classify images in various domains, including scenes and sport events. In [494], Fisher Kernels produce gradient vector representations which are only dependent on the number of parameters of the underlying probability distribution of the samples. Visual vocabularies are generated by GMMs on top of color statistics, SIFT features, and PCA reduction, while SVMs and Sparse Logistic Regression [323] are used for the final discrimination. In [721], local features are obtained via grid-sampled

SIFT, SPM and codebook generation with KMeans, while global responses are obtained via MC-LBP [99], using both the original and a Gabor-filtered input image. Kernel collaborative classification [681] is used to classify scene images.

In the audio domain, additionally to aforementioned benefits for visual data (i.e. dimensionality reduction, noise filtering and computational efficiency), given the sequential nature and importance of temporal interdependencies in audio, aggregation methods may be used to pool together feature responses that are close to each other in the temporal or frequency domain, or to produce content-based segmentation and tokenization into “acoustic words”.

Regarding specific approaches, in [351] low-level MFCC features are aggregated via modelling with single gaussian distributions, GMMs, or PLSA. SVMs with different distance kernels are examined for audio-based classification of consumer videos. In [310] Latent Dirichlet Allocation (LDA) [66] is used to generate probabilistic topics. “Acoustic words” are modeled from MFCC features and vector quantization with the LBG-VQ algorithm [197]. An SVM with a Bhattacharaya kernel [281] and GMMs are for audio tag classification. The approach in [35] adopts PCA and the Max-Relevance, Min-Redundancy method [492] for feature reduction of multiple audio features, related to spectral, dynamic, harmonic and rhythm characteristics, along with higher order moments. SVMs are subsequently used for genre classification.

In summary, this section covered approaches that improve low-level representations by pooling, combining and transforming them into mid-level information. In the next section we conclude the examination of knowledge-agnostic approaches by considering final category formulated in section 2.2: deep representations, i.e. approaches that employ hierarchical feature sets and non-linearity to learn high-level representations in an end-to-end fashion.

2.4 Deep representation methods

The approaches covered up to this point rely on the use of preconfigured templates (e.g. kernels) interpolated to the local neighbourhood of available training data points, as well as manipulations of their responses in fixed, pre-configured steps. In this section, we focus on “deep” feature extractors, which aim to learn a multiplicity of useful *feature hierarchies* from a set of training data [52].

2.4.1 Overview

The typical representatives of architectures that produce deep features are graph-based computational models, such as the biologically inspired artificial neural networks [267, 52]. In this context, *deep* refers to the size of the directed acyclic graph that implements the computation. The hierarchy is composed through multiple steps of non-linear combinations, automatically forming and discovering complex representation functions during

training. Examples of deep features in, e.g., visual data, typically range from pixel-level primitives (oriented edges, blobs and textures) to small object parts and high level concepts in the training data, e.g. “car”, “person” or “sky”. A similar progression of primitive information to higher-level abstractions can be observed in neural modelling for text (e.g. from subword, to passage and document-level conceptual information) and audio (e.g. from short utterances and auditory primitives to identifiable speech / acoustic events). Learning a large set of abstract related concepts could enable generalizations similar to ones performed by humans, as new unseen concepts could potentially be described by meaningful combinations of already learned building blocks [52, 49]. Overall, the interpretability and explainability of deep models and the resulting representations is an open and active area of research [25, 43, 708].

Deep features exhibit superior expressive power and compactness compared to shallow architectures and local estimators for machine learning tasks [47]. Further, deep representation hierarchies can take advantage of unsupervised pretraining [53], supervised fine-tuning and transfer learning [719], e.g. through techniques like Restricted Boltzmann Machines [249], Autoencoders [54], distributional learning [440] and others. Fitted model components can be used to initialize deep neural networks prior to supervised training / fine-tuning on downstream tasks, or be used directly as standalone features [250, 47, 505]. Thus, the nature of deep representations encourages re-use of learned input mappings across tasks and datasets in a straightforward way, which has been investigated and demonstrated for a variety of tasks, domains and modalities [691, 515]. Moreover, in low-level and aggregation-based approaches, the distinction between the feature extraction / manipulation and the learning model is well-defined. In contrast, the computational steps (i.e., layers) of deep models can be thought of as altering the input data through a series of (learnable) transformations, producing ever more complex, abstract and high-level representations. In this paradigm, the line between learning the features and learning the task is often blurred; the entire pipeline is an ensemble that acts both as a feature transformation engine, representation learner and prediction machine, tuning and optimizing intermediate components into final predictions in a direct, *end-to-end* fashion. Contrary to aggregation-based methods, this enables the design of efficient representation learning architectures with fewer hard-coded, performance-critical parameters.

In summary, deep models result in more efficient learning, with respect to the trade-off between performance gains and feature reusability versus computational resources and number of model parameters. In the next paragraphs we will cover specific studies and related work over text, image and audio data that adopt such approaches.

2.4.2 Approaches

Here we cover related work utilizing deep representations for classification tasks on different modalities. As in preceding sections, we provide a summary of the works examined in table 2.1.

In text, the generation of deep transferable features usually utilizes neural language mod-

els (NNLMs), i.e. structures that attempt to capture grammatical and linguistic rules of the input language in its internal organization [82, 24, 236, 51]. This is usually achieved by adopting the distributional hypothesis [235] and learning the feature hierarchy as a function of input token co-occurrence statistics over large amounts of training data. Early approaches had been focused on learning single-layer word embeddings with no established method for generation and fine-tuning of transferable representations, despite the observed benefits of unsupervised pretraining [164] observed in other media. Despite this, deep learning approaches can outperform simpler models on producing rich representations, as comparative studies have indicated. For example, Baroni et. al [37] compare neural embeddings with low-level term-based, as well as post-processed vector-space representations via aggregation and transformation methods. Intra-evaluation of the generated features on semantic relatedness, synonym detection and concept categorization provides evidence that the neural approaches produce rich, expressive embeddings, outperforming count-based counterparts on every examined task.

A statistical NNLM is used in [682] – there, TF-IDF - SVM baselines are compared to using concatenated word embeddings in conjunction with a network with convolutional and pooling layers, for the task of classifying tweets related to elections. The Word2vec algorithm [440] is used to train the word embeddings, over Wikipedia and Twitter micropost datasets. In [103] four word embedding approaches are evaluated in multiple classification tasks. Namely, the approaches examined include two pairwise n-gram embeddings (SENNNA [118] and the modified version in [624]), the embeddings produced by the Hierarchical Log-Bilinear (HLBL) statistical model [445] and the multisense cluster-based word embeddings from [262]. The tasks examined included sentiment analysis as well as gender, plurality and synonym/antonym classification, using Logistic Regression and RBF kernel SVMs. The work in [382] uses LDA [66] to build word and topic embeddings, using Skip-gram [440] vectors as seed word vectors. The authors explore variants that embed topics, word-topic pairs, or the concatenation of the generated topic and word embedding vectors. Aggregating to document embeddings, the authors evaluate the approach on news categorization with a linear classifier. A convolutional neural model is used in [118] in a multitask learning scenario (including semantic role labeling, language modeling, named-entity recognition and others), using convolutional, max-pooling and fully connected layers, in conjunction with lookup tables. The same model is trained in a semi-supervised manner, with all tasks using labelled data except from the language modelling objective. The architecture is evaluated on multiple tasks, including binary and multiclass classification (semantically related word prediction and POS tagging). Convolution is also used in [337], where a bi-directional recurrence structure is employed, argued to counter the bias of Recurrent Neural Network [161] (RNN) models towards latter sequence elements, as well as the high time complexity of recursive NNs [587]. The model utilizes two context vectors to encapsulate the left and right word sequence contexts around a center word, with the output of the recurrence being filtered into a fixed-length representation of salient features with a max-pooling layer. The model is tested by fine-tuning the SkipGram model for classifying news, scientific and web content documents.

Additionally, in [404], a probabilistic NLM [51] learns representations from existing infor-

mation in a term-document matrix structure, using a statistical log-linear document model to represent documents as a mixture of word distributions. The approach resembles generation of word-topic LDA associations, however it models real-valued (i.e., not restricted in the unit simplex) word vectors and does not include a topic modelling objective. An evaluation is performed on document sentiment classification and subjectivity detection of movie reviews using a linear SVM over averaged word vectors. The work in [347] produces document embeddings using a paragraph-level context, using an approach similar to the word2vec algorithm [440]. The paragraph vector is either concatenated to all contexts in the same paragraph (i.e. acting as a shared common word), or used as the center word for prediction of the context. The vectors are used on sentiment analysis on movie reviews, using a shallow neural network as the sentiment predictor. Recent approaches utilize self-attention and transformer constructs to produce transferable representations. In [604], the authors propose multiple avenues towards fine-tuning the BERT model [148] for classification, including additional pre-training, single-task and multi-task fine-tuning with varying learning rates per layer in the transformer. The resulting model is evaluated on sentiment analysis on reviews over multiple domains, as well as topic and question classification.

In images, the large semantic gap between raw image pixels and conceptual information has encouraged the design of deep architectures of hierarchical features. Early approaches such as Alexnet [324] applied repeated layers of convolution, pooling and normalization operators, followed by dropout-regularized [594] dense (i.e. fully-connected) connections and softmax normalization. The network was used for classifying images to Imagenet classes [535] with a neural fully-connected layer with softmax normalization. In [580], the authors increase the depth of the Alexnet architecture, applied on the same classification setting. The produced models are scaled up to 19 layers by featuring smaller convolution kernels in the early steps of the graph. Going deeper, the GoogleNet model [610] employed “inception modules” for the same task and classification setting as the previous works. Inception modules correspond to designed constructs that split incoming inputs into multiple branches of smaller embeddings via dimensionality reduction operations (1-by-1 convolutions and pooling). The responses of these modules are subsequently filtered and re-concatenated. These building blocks form the computational chain, supplemented with fully-connected components into networks employed for image recognition tasks, with scaling and regularization improvements are applied in future works [611, 609]. In order to tackle the problem of the vanishing gradient during gradient-based learning [251] as well as mitigate representation redundancy in the feature hierarchy, the authors of the ResNet model [239] utilize multi-branching. This is implemented by introducing residual connections in the network graph, in the form of binary splits where one is the identity mapping. This allows for easier and effective training of very large models compared to previous approaches, as illustrated on evaluation in multiple large-scale classification tasks. The DenseNet model [263] expands the residual connection scheme by building “densely connected” sequences: the proposed topology links a layer output to the input of all subsequent layers and vice versa. Additionally, instead of performing an identity additive mapping (i.e. like ResNets [239]), feature concatenation is used. The approach is effective for a wide range of image classification tasks – additionally, the model

requires fewer parameters than ResNets, which is achieved by utilizing compression, bottlenecking and growth control techniques in the architecture.

Regarding deep approaches for handling audio, an established technique has been to convert the audio content into the visual modality and apply an image classification pipeline on the resulting images. A popular approach towards this is to convert the audio into a spectrogram image, a two-dimensional time and frequency representation, produced by applying short-time Fourier transform [566] on segmented audio clips. For example, in [350] convolutional deep belief networks [247] are employed to learn deep features for phoneme representation in an unsupervised manner. The generated audio features are used for a variety of music and speaker classification tasks, using SVM, k-NN, GDA classifiers, as well as ensemble methods. A similar approach is undertaken in [705] for genre classification tasks, where convolutional, pooling and projection layers are used to generate deep audio features, mimicking the function of the visual cortex [267]. A linear ridge regression classifier is used to produce the final prediction. In [266], convolutional networks are used with a two-stage learning procedure over audio spectrograms: first, unsupervised training with an encoder-decoder reconstruction task captures the structure of the spectrogram at multiple scales. Subsequently, a fully-connected layer combines the convolutional pooled output into a feature vector, fitted in a semi-supervised manner. Both workflows are trained jointly, with the output logits being fed to an SVM for emotion classification. The approach in [107] uses deconvolution [703] to produce audio corresponding to weights of trained CNN networks, with the goal of improving the explainability of learned features. They utilize an architecture of convolution, pooling and fully-connected layers on audio spectrograms, evaluated on music genre classification. In [242], the authors perform a large-scale evaluation of popular convolutional deep architectures from the image classification domain [324, 580, 610, 239]. The authors generate audio spectrograms and apply each architecture for audio event categorization, additionally varying the amount of data and the labelset size for each experiment. All visual models considerably outperform a fully-connected neural network on raw spectrogram input features, with ResNet and the Inception [611] architecture performing best.

2.5 Method Comparison

In the previous sections we presented three general approaches for constructing content-based representations, in terms of the richness of semantics encapsulated in the output features. Here we provide a comparative summary between them, considering some common desiderata for representations [47]. The comparison is facilitated in table 2.2, where rows list useful attributes representation systems and columns consist of the three paradigms investigated in this chapter. A green checkmark indicates that the attribute holds for the corresponding paradigm, a red X denotes that it does not, while yellow question marks show that it is not clear or straightforward whether the attribute characterizes the representation paradigm. Note that this comparison is based on traits of paradigms in general, with edge cases, grey areas and exceptions being unavoidable.

Desired Attributes	Representation Paradigm		
	Low-level	Aggregation	Deep
high-level semantics	X	?	✓
explainable	✓	?	X
data-driven / learned	X	?	✓
low-dimensional / space-efficient	?	✓	✓
data efficient / lean	✓	✓	X
computationally efficient	?	X	X

Table 2.2: Comparison between representation approaches. Green checkmarks, red X's and yellow question marks indicate whether the desired attribute in the corresponding row generally holds, does not hold, or it is unclear whether it applies, respectively, for the representation generation paradigm in the corresponding column.

- Regarding low-level and template matching approaches (first column), it generally holds that the representations they generate are explainable (i.e., each feature coordinate has a clear, non-ambiguous meaning, which can be discovered by referring to the generation algorithm) and this generation does not require large amounts of data. On the other hand, they heavily rely on handcrafted features and feature engineering, which often requires expert knowledge and familiarity to the domain of application. Additionally, methods in the low-level paradigm produce features that are comparatively lacking in richness of encapsulated semantics and high-level conceptual information. Further, such methods may need to build very large feature spaces to arrive at an adequate expressive power to facilitate efficient classification, leading to high-dimensional representations and, as a result, large requirements for computational resources.
- Aggregation-based approaches (second column) generally successfully deal with dimensionality issues, having the ability building space-efficient representations that can often be configured to a desired size for the needs of specific tasks. These methods use low-level / template matching features as inputs, generally retaining the low requirements for amount of data necessary to build efficient aggregations / transformations, but with the additional computational step leading to an increased need for compute power. Aggregation methods often generate distributed feature spaces, which harms explainability but improves the expressive power of the final representation. Finally, most such approaches employ some degree of unsupervised feature learning, but do so by using fixed, preconfigured rules and analytic solutions.
- Deep representations fully utilize representation learning by accumulating improvements learned from data in an incremental, partially stochastic manner. This generally endows deep representations with semantically rich, distributed, compact features. However, this comes at the cost of creating black box-like feature extractors with very low explainability. Additionally, the reliance of these methods on distributional, data-driven operation renders them highly demanding with respect to the

required amount of data and computational resources.

These observations illustrate that each approach has clear advantages and disadvantages with no definite one-size-fits-all approach, i.e. we have indications for a no free lunch theorem [7] for representations approaches in classification. However, we would suggest that the evolution of content-based approaches from Low-level, to Aggregation-based, to Deep Representations reflects a trend towards representation of increasing richness in semantic content and conceptual information.

2.6 Conclusion

In this chapter we presented content-based representation methods for machine learning tasks over text, image and audio modalities. In summary:

- The covered work was organized into three broad meaningful paradigms, with respect to the sophistication of the data representation paradigm employed.
- The first category covered template-matching methods that extracted low-level information in the image. These methods heavily rely on handcrafted feature engineering and yield global (spanning the entire input instance) or local (pertaining to identified regions of interest) responses.
- The next category included approaches that combine, transform and/or post-process results from the aforementioned low-level features, aggregating their responses to “mid-level”, distributed representations, increasing their information content and generality of encapsulated information.
- In the third category, we investigated deep representation systems, i.e. methods that build hierarchical feature sets via representation learning. These approaches construct classifiers and feature extractors in an end-to-end fashion, utilizing joint task / representation optimization and exhibiting strong transferability of learned features.
- Covered studies were cataloged and summarized in tabular format in order to facilitate comparison across approaches in the literature (i.e. in terms of representation and knowledge injection methodologies, modality, classifiers, metrics, etc.) and available knowledge resources (e.g. providing information about the information unit and relations encapsulated in the resource, compilation type, format / availability, language, etc.).

This overview showcases the multiple approaches and the magnitude of research effort for arriving at rich feature sets for classification tasks. This pursuit bears similarities across modalities, illustrated by the category paradigms identified above. In the following chapter we continue with a closer look on novel applications, extensions and/or modifications of such representation methods on specific tasks of interest (e.g. classification and summarization applications, for text, image and audio data). These proposals are investigated

and evaluated in targeted studies, under the goal of studying the contribution of representations approaches of varying levels of output feature richness, over different machine learning problems.

3. NOVEL APPLICATIONS AND STUDIES USING CONTENT-BASED REPRESENTATIONS

Having conducted a large-scale investigation, cataloguing and comparison of representation approaches for different data modalities and tasks, this chapter focuses on application of resulting insights, findings and methods in different machine learning tasks.

3.1 Hate Speech Detection of Social Media Content

In this section we apply literature approaches over text representations to the task of Hate Speech Detection.

3.1.1 Introduction and Overview

Hate Speech is a common affliction in modern society. Nowadays, people can come across Hate Speech content even more easily through social media platforms, websites and forums containing user-created content. The increase of the use of social media gives individuals the opportunity to easily spread hateful content and reach a large number of people than ever before. On the other hand, social media platforms like Facebook or Twitter want to both comply with legislation against Hate Speech and improve user experience. Therefore, they need to track and remove Hate Speech content from their websites efficiently.

Due to the large amount of data transmitted through these platforms, delegating such a task to humans is extremely inefficient. A usual compromise is to rely on user reports in order to review only the reported posts and comments. This is also ineffective, since it relies on the users' subjectivity and trustworthiness, as well as their ability to thoroughly track and flag such content. Due to all the above, the development of automated tools to detect Hate Speech content is deemed necessary. The goal of this work is: (i) to study different text representations and classification algorithms in the task of Hate Speech detection; (ii) evaluate whether the n-gram graphs representation [204] can constitute a rich/deep feature set (as e.g. in [482]) for the given task.

The structure of this study is as follows. In section 3.1.2 we define the hate speech detection problem, while in section 3.1.3 we discuss related work. We overview our study approach and elaborate on the proposed method in section 3.1.4. We then experimentally evaluate the performance of different approaches in Section 3.1.5, concluding this work in Section 3.1.6, by summarizing the findings and proposing future work.

3.1.2 Problem Definition

The first step to Hate Speech detection is to provide a clear and concise definition of Hate Speech. This is important especially during the manual compilation of Hate Speech detection datasets, where human annotators are involved. In their work, the authors of [331] have asked three students of different race and same age and gender to annotate whether a tweet contained Hate Speech or not, as well as the degree of its offensiveness. The agreement was only 33%, showing that Hate Speech detection can be highly subjective and dependent on the educational and/or cultural background of the annotator. Thus, an unambiguous definition is necessary to eliminate any such personal bias in the annotation process.

Usually, Hate Speech is associated with insults or threats. Following the definition provided by [81], “it covers all forms of expressions that spread, incite, promote or justify racial hatred, xenophobia, antisemitism or other forms of hatred based on intolerance”. Moreover, it can be “insulting, degrading, defaming, negatively stereotyping or inciting hatred, discrimination or violence against people in virtue of their race, ethnicity, nationality, religion, sexual orientation, disability, gender identity”. However, we cannot disregard that Hate Speech can be also expressed by statements promoting superiority of one group of people against another, or by expressing stereotypes against a group of people.

The goal of a Hate Speech Detection model is, given an input text T , to output True, if T contains Hate Speech and False otherwise. Modeling the task as a binary classification problem, the detector is built by learning from a training set and is subsequently evaluated on unseen data. Specifically, the input is transformed to a machine-readable format via a text representation method, which ideally captures and retains informative characteristics in the input text. The representation data is fed to a machine learning algorithm that assigns the input to one of the two classes, with a certain confidence. During the training phase, this discrimination information is used to construct the classifier. The classifier is then applied on data not encountered during training, in order to measure its generalization ability.

In this study, we focus on user-generated texts from social media platforms, specifically Twitter posts. We evaluate the performance of several established text representations (e.g. Bag of words, word embeddings) and classification algorithms. We also investigate the contribution of the graph-based n-gram graph features to the Hate Speech classification process. Moreover, we examine whether a combination of deep features (such as n-gram graphs) and shallow features (such as Bag of Words) can provide top performance in the Hate Speech detection task.

3.1.3 Related Work

In this section, we provide a short review of the related work, not only for Hate Speech detection, but for similar tasks as well. Examples of such tasks can be found in [451] where the authors aim to identify which users express Hate Speech more often, while [676] detect

and delete hateful content in a comment, making sure what is left has correct syntax. The latter is a demanding task which requires the precise identification of grammatical relations and typed dependencies among words of a sentence. Their proposed method results have 90.94% agreement with the manual filtering results.

Automatic Hate Speech detection is usually modeled as a Binary Classification. However, multi-class classification can be applied to identify the specific kind of Hate Speech (e.g. racism, sexism etc) [30, 485]. One other useful task is the detection of the specific words or phrases that are offensive or promote hatred, investigated in [657].

3.1.3.1 Text representations for Hate Speech

In this work we focus on representations, i.e. the mapping of written human language into a collection of useful features in a form that is understandable by a computer and, by extension, a Hate Speech Detection model. Below we overview a number of different representations used within this domain.

A very popular representation approach is the Bag of Words (BOW) [331, 73, 30] model, a Vector Space Model extensively used in Natural Language Processing and document classification. In BOW, the text is segmented to words, followed by the construction of a histogram of (possibly weighted) word frequencies. Since BOW discards word order, syntactic, semantic and grammatical information, it is commonly used as a baseline in NLP tasks. An extension of the BOW is the Bag of N-grams [470, 331, 451, 138, 659], which replaces the unit of interest in BOW from words to n contiguous tokens. A token is usually a word or a character in the text, giving rise to word n-gram and character n-gram models. Due to the contiguity consideration, n-gram bags retain local spatial and order information.

The authors in [138] claim that lexicon detection methods alone are inadequate in distinguishing between Hate Speech and Offensive Language, counter-proposing n-gram bags with TF-IDF weighting along with a sentiment lexicon, classified with L2 regularized Logistic Regression [424]. On the other hand, [30] use character n-grams, BOW and TF-IDF features as a baseline, proposing word embeddings from GloVe¹. In [485] the authors use character and word CNNs as well a hybrid CNN model to classify sexist and racist Twitter content. They compare multi-class detection with a coarse-to-fine two-step classification process, achieving similar results with both approaches. There is also a variety of other features used such as word or paragraph embeddings ([152], [657], [30]), LDA and Brown Clustering ([543], [671], [657], [659]), sentiment analysis([214], [138]), lexicons and dictionaries ([214], [577], [140] etc) and POS tags([470], [676], [543] etc).

¹<https://nlp.stanford.edu/projects/glove/>

3.1.3.2 Classification approaches

Regarding classification algorithms, SVM [126], Logistic Regression (LR) and Naive Bayes (NB) are the most widely used (e.g. [657, 543, 138, 152] etc). In [660] and [671], the authors use a bootstrapping approach to aid the training process via data generation. This approach was used as a semi-supervised learning process to generate additional data automatically or create hatred lexical resources. The authors of [671] use the Map-Reduce framework in Hadoop to collect tweets automatically from users that are known to use offensive language, and a bootstrapping method to extract topics from tweets.

Other algorithms used are Decision Trees and Random Forests (RF) ([138, 73, 671]), while [30] and [140] have used Deep Learning approaches via LSTM networks. Specifically, [30] use CNN, LSTM and FastText, i.e. a model that is represented by average word vectors similar to BOW, which are updated through backpropagation. The LSTM model achieved the best performance with 0.93 F-Measure, used to train a GBDT (Gradient Boosted Decision Trees) classifier. In [138], the authors use several classification algorithms such as regularized LR, NB, Decision Trees, RF and Linear SVM, with L2-regularized LR outperforming other approaches in terms of F-score.

For more information, the survey of [557] provides a detailed analysis of detector components used for Hate Speech detection and similar tasks.

3.1.4 Study and Proposed Method

In this section we will describe the text representations and classification components used in our implementations of a Hate Speech Detection pipeline. We have used a variety of different text representations, i.e. bag of words, embeddings, n-grams and n-gram graphs and tested these representations with multiple classification algorithms. We have implemented feature extraction in Java and used both Weka and scikit-learn (sklearn) to implement classification algorithms. For artificial neural networks (ANNs), we have used sklearn and Keras frameworks. Our model can be found in our GitHub repository ².

3.1.4.1 Text representations

In order to discard noise and useless artifacts we apply standard preprocessing to each tweet. First, we remove all URLs, mentions (e.g. @username), RT (Retweets) and hashtags (e.g. words starting with #), as well as punctuation, focusing on the text portion of the tweet. Second, we convert tweets to lowercase and remove common English stopwords using a predefined collection ³.

After preprocessing, we apply a variety of representations, starting with the Bag of Words (BOW) model. This representation results in a high dimensional vector, containing all

²<https://github.com/cthem/hate-speech-detection>

³<https://github.com/igorbrigadir/stopwords>

encountered words, requiring a significant amount of time in order to process each text. In order to reduce time and space complexity, we limit the number of words of interest to keywords from HateBase⁴ [138].

Moreover, we have used additional bag models, with respect to word and character n-grams. In order to guarantee a common bag feature vector dimension across texts, we pre-compute all n-grams that appear in the dataset, resulting in a sparse and high-dimensional vector. Similarly to the BOW features, in order to reduce time and space complexity, it is necessary to reduce the vector space. Therefore, we keep only the 100 most frequent n-grams features, discarding the rest. Unfortunately, as we will illustrate in the experiments, this decision resulted in highly sparse vectors and, thus, reduced the efficiency of those features.

Furthermore, we have used GloVe word embeddings [493] to represent the words of each tweet, mapping each word to a 50-dimensional real vector and arriving at a single tweet vector representation via mean averaging. Words missing from the GloVe mapping were discarded.

Expanding the use of n-grams, we examine whether n-gram graphs (NGGs) [199, 208] can have a significant contribution in detecting Hate Speech. NGGs are a graph-based text representation method that captures both frequency and local context information from text n-grams (as opposed to frequency-only statistics that bag models aggregate). This enables NGGs to differentiate between morphologically similar but semantically different words, since the information kept is not only the specific n-gram but also its context (neighboring n-grams). The graph is constructed with n-grams as nodes and local co-occurrence information embedded in the edge weights, with comparisons defined via graph-based similarity measures [199]. NGGs can operate with word or character n-grams – in this work we employ the latter version, which has been known to be resilient to social media text noise [482, 208].

During training, we construct a representative category graph (RCG) for each category in the problem (e.g. “Hate Speech” or “Clean”), aggregating all training instances per category to a single NGG. We then compare the NGG of each instance to each RCG, extracting a score expressing the degree that the instance belongs to that class – for this, we use the NVS measure [199], which produces a similarity score between the instance and category NGGs. After this process completes, we end up with similarity-based, n -dimensional model vector features for each instance – where n is the number of possible classes. We note that we use 90% of the training instances to build the RCGs, in order to avoid overfitting of our model: in short, using all training instances would result in very high instance-RCG similarities during training. Since we use the resulting model vectors as inputs to a classification phase in the next step, the above approach would introduce extreme overfit to the classifier, biasing it towards expecting perfect similarity scores in cases of an instance belonging to a class, a scenario which of course rarely – if ever – happens with real world data.

In addition, we produce sentiment, syntax and spelling features. Sentiment analysis could

⁴<https://github.com/t-davidson/hate-speech-and-offensive-language>

be a meaningful feature, since hatred is related with a negative polarity. For sentiment and syntax feature extraction we use the Stanford NLP Parser⁵. This tool performs sentiment extraction of the longest phrase tracked in the input and additionally can be used to provide a syntactic score with syntax trees, corresponding the best attained score for the entire tweet.

Finally, a spelling feature was constructed to examine whether Hate Speech is correlated to the user's proficiency in writing. We have used an English dictionary to collect all English words with correct spelling and, then, for each word in a tweet, we have calculated its edit distance from each word in the dictionary, keeping the smallest value (i.e. the distance from the best match). The final feature kept was the average edit distance for the entire post, with its value being close to 0 for tweets with the majority of words correctly spelled. At the end of this process, we obtain a 3-dimensional vector, each coordinate corresponding to the sentiment, syntax and spelling scores of the text.

3.1.4.2 Classification Methods

Generated features are fed to a classifier that decides the presence of Hate Speech content. We use a variety of classification models, as outlined below.

Naive Bayes (NB) [537] is a simple probabilistic classifier, based on Bayesian statistics. NB makes the strong assumption that instance features are independent from one another, but yields performance comparable to far more complicated classifiers – this is why it commonly serves as a baseline for various machine learning tasks [357]. Additionally, the independence assumption simplifies the learning process, reducing it to the model learning the attributes separately, vastly reducing time complexity on large datasets.

Logistic Regression (LR) [429] is another statistical model commonly applied as a baseline in binary classification tasks. It produces a prediction via a linear combination of the input with a set of weights, passed through a logistic function which squeezes scores in the range between 0 and 1, i.e. thus producing binary classification labels. Training the model involves discovering optimal values for the weights, usually acquired through a maximum likelihood estimation optimization process.

The K-Nearest Neighbor (KNN) classifier [180] is another popular technique applied to classification. It is a lazy and non-parametric method; no explicit training and generalization is performed prior to a query to the classification system, and no assumption is made pertaining to the probability distribution that the data follows. Inference requires a defined distance measure for comparing two instances, via which closest neighbors are extracted. The labels of these neighbors determine, through voting, the predicted label of a given instance.

The Random Forest (RF) [368] is an ensemble learning technique used for both classification and regression tasks. It combines multiple decision trees during the training phase by bootstrap-aggregated ensemble learning, aiming to alleviate noise and overfitting by

⁵<https://nlp.stanford.edu/software/lex-parser.html>

incorporating multiple weak learners. Compared to decision trees, RF produces a split when a subset of the best predictors is randomly selected from the ensemble.

Artificial Neural Networks (ANNs) are computational graphs inspired by the biological nervous systems. They are composed of a large number of highly interconnected neurons, usually organized in layers in a feed-forward directed acyclic graph. Similarly to an LR unit, neurons compute the linear combination of their input (including a bias term) and pass the result through a non-linear activation function. Aggregated into an ANN, each neuron computes a specific feature from its input, as dictated by the values of the weights and bias. ANNs are trained with respect to a loss function, which defines an error gradient by which all parameters of the ANN are shifted. With each optimization step, the model moves towards an optimum parameter configuration. The gradient with respect to all network parameters is computed by the back-propagation method. In our case, we have used an ANN composed of 3 hidden layers with dropout regularization.

3.1.5 Experiments and Results

In this section, we present the experimental setting used to answer the following:

- Which features have the best performance?
- Does feature combination improve performance?
- Do NGGs have significant / comparable performance to BOW or word embeddings despite being represented by low dimensional vectors?
- Are there classifiers performing statistically significantly better than others? Is the selection of features or classifiers more significant, when determining the pipeline for Hate Speech detection?

In the following paragraphs, we elaborate on the datasets utilized, present experimental and statistical significance results and discuss our findings.

3.1.5.1 Datasets and Experimental Setup

We use the datasets provided by [660]⁶ and [138]⁷. We will refer to the first dataset as RS (racism and sexism detection) and to the second as HSOL (distinguish Hate Speech from Offensive Language). In both works, the authors perform a multi-class classification task against the corpora. In [660], their goal is to distinguish different kinds of Hate Speech, i.e. racism and sexism, and therefore the possible classes in RS are Racist, Sexist or None. In [138], the annotated classes are Hate Speech, Offensive Language or Clean.

⁶<https://github.com/ZeerakW/hatespeech>

⁷<https://github.com/t-davidson/hate-speech-and-offensive-language>

Given the multi-class nature of these datasets, we combine them into a single dataset, keeping only instances labeled Hate Speech and Clean in the original. We use the combined (RS + HSOL) dataset to evaluate our model implementations on the binary classification task. Furthermore, we run multi-class experiments on the original datasets for completeness, the results of which are omitted due to space limitations, but are available upon request.

We perform three stages of experiments. First, we run a preliminary evaluation on each feature separately, to assess its performance. Secondly, we evaluate the performance of concatenated feature vectors, in three different combinations: 1) the top individually performing features by a significant margin (best), 2) all features all and 3) vector-based features (vector), i.e. excluding NGGs. Via the latter two scenarios, we investigate whether NGGs can achieve comparable performance to vector-based features of much higher dimensionality.

Given the imbalanced dataset used (24463 Hate Speech and 14548 clean samples), we report performance in both macro and micro F-measure. Finally, we evaluate (with statistical significance testing) the performance difference between run components, through a series of ANOVA and Tukey HSD test evaluations.

3.1.5.2 Results

Here we provide the main experimental results of our described in the previous section, presented in micro/macro F-measure scores. More detailed results, including multi-class classification are omitted due to space limitations but are available upon request.

Firstly, to answer the question on the value of different feature types, we perform individual runs which designate BOW, glove embeddings and NGG as the top performers, with the remaining features (namely sentiment, spelling / syntax analysis and n-grams) performing significantly worse. All approaches however surpass a baseline performance in terms of a naive majority-class classifier (scoring 0.382/0.473, in terms of macro and micro F-measure respectively) and are described below. Sentiment, spelling and syntax features proved to be insufficient information sources to the Hate Speech detection classifiers when used separately – not surprisingly, since they produce one-dimensional features. The best performers are syntax with NNs in terms of micro F-measure (0.633) and spelling with NNs in terms of macro F-measure (0.566). In contrast to n-gram graph similarity-based features perform close to the best performing BOW configuration (cf. Table 3.1), having just one additional dimension. This implies that appropriate, deep / rich features can still offer significant information, despite the low dimensionality. NGG-based features appear to have this quality, as illustrated by the results. Finally, N-grams were severely affected by the top-100 token truncation. The best character n-gram model achieves macro/micro F-Measure scores of 0.507/0.603 with NN classification and the best word n-gram model 0.493/0.627 with KNN and NN classifiers.

The results of the top individually performing features, in terms of micro / macro average F-Measure, are presented in the left half of table 3.1. **Bold** values represent column-wise

Table 3.1: Average micro & macro F-Measure for NGG, BOW and GloVe features (left) and the “best”, “vector” and “all” feature combinations (right).

feature	classifier	macrof	microf	combo	classifiers	macrof	microf
NGG	KNN	0.712	0.736	best	KNN	0.810	0.820
	LR	0.712	0.739		LR	0.819	0.831
	NB	0.678	0.713		NB	0.632	0.667
	NN_ke	<u>0.718</u>	0.727		NN_ke	0.807	0.819
	NN_sk	0.716	<u>0.740</u>		NN_sk	0.819	0.831
	RF	0.699	0.726		RF	0.734	0.759
BOW	KNN	0.787	0.763	all	KNN	0.497	0.569
	LR	0.808	<u>0.776</u>		LR	0.760	0.772
	NB	0.629	0.665		NB	<u>0.795</u>	<u>0.792</u>
	NN_ke	<u>0.808</u>	<u>0.776</u>		NN_ke	0.537	0.629
	NN_sk	<u>0.808</u>	<u>0.776</u>		NN_sk	0.664	0.678
	RF	0.807	0.776		RF	0.700	0.731
glove	KNN	0.741	0.765	vector	KNN	0.497	0.569
	LR	0.749	0.769		LR	0.745	0.756
	NB	0.715	0.726		NB	<u>0.787</u>	<u>0.783</u>
	NN_ke	<u>0.774</u>	0.788		NN_ke	0.592	0.640
	NN_sk	<u>0.786</u>	0.800		NN_sk	0.669	0.675
	RF	0.731	0.755		RF	0.727	0.742

maxima, while underlined ones depict maxima in the left column category (e.g. feature type, in this case). “NN_ke” and “NN_sk” represent the keras and sklearn neural network implementations, respectively. We can observe that the best performer is BOW with either LR or NNs, followed by word embeddings with NN classification. NGGs have a slightly worse performance, which can be attributed to the severely shorter (2D) feature vector it utilizes. On the other hand, BOW features are 1000-dimensional vectors. Compared to NGGs, this corresponds to a 500-fold dimension increase, with a 9.0% micro F-measure performance gain.

Subsequently, we test the question on whether the combination of features achieve a better performance than individual features. The results are illustrated in the right half of Table 3.1. First, the best combination that involves NGG, BOW and GloVe features is, not surprisingly, the top performer, with LR and NN-sklearn obtaining the best performance. The all configuration follows with NB achieving macro/micro F-scores of 0.795 and 0.792 respectively. This shows that the additional features introduced significant amounts of noise, enough to reduce performance by canceling out any potential information the extra features might have provided. Finally, the vector combination achieves the worst performance: 0.787 and 0.783 in macro/micro F-measure. This is testament to the added value NGGs contribute to the feature pool, reinforced by the individual scores of the other vector-based approaches.

Apart from experiments in the binary Hate Speech classification on the combined dataset,

Table 3.2: ANOVA results with respect to feature and classifier selection, in terms of macro F-measure (left) and micro-Fmeasure (right).

parameter	Pr(>F) (macrof)	Pr(>F) (microf)
features	< 2e-16	< 2e-16
classifiers	2.77e-05	8.65e-08

we have tested our classification models in multi-class classification, using the original RS and HSOL datasets. In RS, our best score was achieved with the `all` combination and the RF classifier with a micro F-Measure of 0.696. For the HSOL dataset, we achieved a micro F-Measure of 0.855, using the best feature combination and the LR classifier.

3.1.5.3 Significance testing

In table 3.2 we present ANOVA results with respect to feature extractors and classifiers, under macro and micro F-measure scores. For both metrics, the selection of both features and classifiers is statistically significant with a confidence level greater than 99.9%. We continue by performing a set of Tukey’s Honest Significance Difference test experiments in table 3.3, depicting each statistically different group as a letter. In the upper part we present results between feature combination groups (“a” to “d”), where the best combination is significantly different by the similar `all` and `vector` combinations by a large margin, as expected. The middle part compares individual features (grouped from “a” to “g”), where GloVe, BoW and NGGs are assigned to neighbouring groups and arise the most significant features, with the other approaches having a large significance margin from them. Spelling and syntax features are grouped together, as well as the n-gram approaches. Finally, the lower part of the table examines classifier groups (“a” to “c”). Here LR leads the ranking, followed by groups with the ANNs approaches, the NB and RF, and the KNN method.

3.1.5.4 Discussion

The results and statistical tests on our work showcase the BOW, GloVe embeddings and the NGG model as the top performing feature-related configurations. BOW and GloVe score best in terms of micro and macro F-measure respectively, with NGG close behind, despite the extreme dimensionality reduction incurred by the model vector representation of graph similarities. The combination of the top performing features improves the results over individual ones, with 0.831 micro F-Measure when employed on an LR classifier or NN-sklearn.

Regarding classification methods, the LR and ANN classifiers perform best when used with our top performing features (separately or combined). Statistical tests show that in both micro and macro F-Measure terms, both representation and classification approaches have a significant role in the performance results.

Table 3.3: Tukey’s HSD group test on micro F-Measure between feature combination groups (top), individual features (middle) and classifiers (bottom).

config	micro F-measure	groups
best	0.787	a
all	0.695	cd
vector	0.693	d
glove	0.767	a
BoW	0.755	ab
NGG	0.730	bc
spelling	0.617	e
syntax	0.613	e
c-ngrams	0.574	f
w-ngrams	0.572	f
sentiment	0.500	g
LR	0.689	a
NN_ke	0.670	ab
NN_sk	0.668	ab
NB	0.661	bc
RF	0.655	bc
KNN	0.639	c

Finally, we understand from our study that the contribution of NGGs as a text representation is significant. NGGs do not use domain-specific knowledge (unlike the BOW vectors which use HateBase keywords) nor require prior training on large document collections (unlike word embeddings, which need extensive unsupervised pre-training). In addition, the vector dimension of the NGG-based approach is equal to the number of classes, as opposed to the 1000 and 50-dimensional BOW and embedding vectors, respectively. Despite this low dimensional representation, our empirical evaluation shows that NGGs have a significant contribution to detection performance. Therefore, NGGs can be seen as off-the-shelf rich features that encapsulate useful information in a low dimensional representation, which helps achieve significant performance either when used by itself or in feature combination approaches.

3.1.6 Conclusion and Future Work

In this study, we investigated different text representation techniques and classification algorithms, performing a large number of experimental evaluations on the Hate Speech detection problem. We showed that n-gram graph-based features constitute deep/rich features, with significant contribution to the Hate Speech classification results.

Viable extensions of this work would be a more detailed evaluation of the contribution of word roles (e.g. POS tags) and their combination with improved preprocessing, to avoid possible noise in the related features. Concerning NGGs in Hate Speech detection,

findings of previous work on NGG variations could be applied [622], to represent short texts with only the important n-grams of the text (e.g. through a TF-IDF filtering process and/or a named entity recognizer). The aim is to reduce the complexity and size of the NGGs, while retaining all the useful information. Another avenue of research is the enrichment of deep features with statistical pre-trained models (such as Latent Dirichlet Allocation [66]) or semantic information (e.g. from thesauri) to further improve performance.

3.2 Extractive Summarization of Web Documents

We continue by considering the extractive summarization task; here, we approach summary construction in a sentence-based classification perspective, adopting a topic modelling-inspired representation for mapping text to a vector space.

3.2.1 Introduction and Overview

In recent years, advances in the field of Natural Language Processing (NLP) have revolutionized the way machines are used to interpret human-written text. With the rapid accumulation of publicly available documents, from newspaper articles to social media posts, machine learning methods designed to automate data analysis are urgently needed. A problem that has been relevant since the dawn of NLP is the automatic summary extraction from a large corpus of text. The development of a consistent and time-efficient method of extractive summarization can assist journalists in their day-to-day tasks, as well as provide better tools for information retrieval.

Summaries need to be as brief as possible but must also capture the important elements of a text. This turns out to be a challenging task for any algorithm to carry out, since there is a virtually infinite number of documents that can exist, and each one of them can refer to a unique concept. Natural language is tricky for a computer to model; the absence or presence of a single word can shift the meaning of a whole sentence or even of a whole chapter. On the other hand, some words do not add any value to a sentence, the meaning is still the same even if we ignore them. To make matters even more complex, a word can be crucial for one article but of little importance to another.

Human brains have evolved to effectively detect complex patterns in text, to focus on the most important bits of a text while ignoring those that are less important. For a machine, the importance of a word or a sentence is not obvious, as it needs to be programmed with a built-in way to assess it in any given context. For the purposes of summary extraction, an automatic summarizer needs to be able to compare words, or sentences via computational means, and announce those with the highest scores as the most relevant for a given document. The representation, i.e. the method by which these similarity scores are assigned, is of critical importance to any summary extraction task.

When the representation is selected, the next step is training the model, that is, feeding the sentences represented as numerical sequences, to a machine learning procedure. If

the representation and the dataset are suitable for the goal we are trying to accomplish, we can expect that the model will be able to predict which words or sentences are more important to a given document. Summing up all the sentences that the model considers to be important, results in a summary of the input text.

3.2.2 Related work

3.2.2.1 Topic Modeling

Topics can be viewed as semantic groups that refer to a particular portion of reality. A document can refer to one or more distinct topics, which humans often can easily distinguish. For example, the words "fishing", "boat", "waves", have something in common; they are all affiliated with the sea. We can think of Sea as one topic, which contains these three words. However, topics are not always that identifiable and there can be broader or narrower topics. Resuming the previous example, alternatively, there can exist a topic on fishing, another one on boats and another one on ocean waves. Each one of them contains a number of words that are directly tied to that concept.

As demonstrated, there is no unique way to infer topics from an input document. It depends on the representation, the way that we measure the similarity scores between two words. It only makes sense that if two words are similar, they will have a high chance of belonging to the same topic. This statement derives from the distributional hypothesis in linguistics which proposes that words that occur in similar contexts tend to have similar meanings [235] However, we have to keep in mind that one word can also belong to one or more topics and that the number of topics in a document is also not known.

Topic models can infer topics by observing the distribution of words across documents. This can be accomplished with Latent Dirichlet Allocation (LDA) [597, 65], a generative statistical model that makes the hypothesis that there exists an underlying distribution of words, topics and documents, which generated the input text collection. Using probabilistic topic model jargon, the words of a document are called "observed variables", whereas the variables of the topic structure are called "hidden variables". Using an iterative process, the model estimates the posterior distribution of the hidden variables given the observed variables. However, the vast amount of topic structures that can exist result in exponential complexities of computation. For this reason, sampling-based algorithms have been developed, such as Gibbs sampling.

In Gibbs sampling [597], a Markov chain (i.e., a sequence of random variables, each only dependent on the previous) is constructed, using samples from the distribution of hidden variables. The assignment of words to topics is sampled iteratively until the Markov chain converges to the target distribution. In the beginning of this procedure, each word is randomly assigned to a topic and in each subsequent iteration, the word-topic assignments are re-evaluated, which might result in words passing through multiple topics during the process.

3.2.2.2 Vector Space Models

Vector Space Model (VSM) approaches project the input to an n -dimensional vector representation, where the semantic similarity of the points is determined by their distance (e.g. cosine, euclidean, etc.) in the projected vector space. Feature vector representations are widely used in Machine Learning tasks, e.g. for classification, clustering, etc. of a collection of input items [625].

A popular way to represent a set of documents as feature vectors has been the bag-of-words approach [545], where a sentence can be represented as a vector of word features. Each vector coordinate expresses word statistics, such as frequency or the Term Frequency-Inverse Document Frequency (TF-IDF) [293] value of a given word in the source texts. By mapping a word to its TF-IDF value, words receive a high weight when they often appear in the referenced document, but rarely in other documents of the set. The benefit of this approach is that it suppresses common words that appear in the majority of documents, without containing any semantic value for the task. It has been demonstrated that the approach can result in significant improvements over raw frequency approaches in a variety of information retrieval tasks. [544].

3.2.2.3 Extractive Summarization

In extractive summarization, the summaries produced contain a subset of unmodified sentences contained in the original documents. Consequently, in these approaches, sentences, and not words, consist of the units of feature selection. The pipeline of an extractive text summarizer is formed of three relatively independent tasks [518]:

1. Construction of an intermediate representation of the input text based on the key aspects of the text
2. Scoring the sentences based on the selected representation
3. Selection of the summary consisting of a number of sentences

In [232], the authors define a different division of tasks, which includes a pre-processing and a processing step. The pre-processing step also includes: sentence boundary identification, stop-word elimination, and stemming. During the processing step, weights are assigned to specific sentence features by a feature-wise weighting mechanism, with the top ranked sentences being included in the final summary. In this study, we will follow the paradigm, proposed by [518].

There are two types of representation-based approaches: 1) topic representations and indicator representations. A Topic representation transforms the text into an intermediate form and interprets the topic(s) discussed in the text. The techniques used for this, differ in terms of their complexity, and are divided into frequency-driven approaches, topic word approaches, latent semantic analysis and Bayesian topic models. Indicator representation describes every sentence as a list of formal features (indicators) of importance such as sentence length, position in the document, or having certain phrases; the use of indicators was demonstrated by [274].

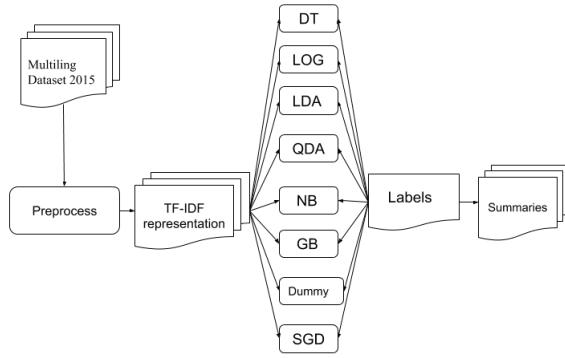


Figure 3.1: The pipeline for the TF-IDF-based extractive summarization.

In contrast to bag-of-words representations that suffer from the curse of dimensionality [45], more sophisticated recent approaches produce sentence vectors in a lower dimensional space, such as a latent-topic space. Many such methods utilize topic clusters in order to locate the centroids (or medoids in non-euclidean spaces) that best represent the sentences in the topics. Then the score of each sentence is assigned in respect to its distance from the clusters' representatives. For example, [618] used a graph-based procedure where each node of the graph represents a sentence and the edges' weights reflect the similarity between the connected nodes. Next, a PageRank / TextRank algorithm is applied to extract the sentence representatives based on the graph centrality. In another topic-based approach, featured by [640] Principal Component Analysis (PCA) was used to project the sentences into a lower-dimension space. The principal components are then evaluated and the sentences with the highest scores get selected to appear in the summary.

Our contribution strives to address limitations with the majority of the existing topic-based summarization methods. First, they work directly in the sentence space and the term-topic information embedded in the sentences is ignored.

In this study, we combine the simplicity of word-level approaches with the power of probabilistic topic models; instead of limiting word information to a single value (e.g. frequency or TF-IDF weight), we model sentences with word-level topic assignments. This approach is supported by a clear and rigorous probabilistic interpretation (rather than some ad-hoc sentence-level aggregation of a multitude of unrelated scores) and produces rich, semantic sentence-level representations.

3.2.3 Proposed Method

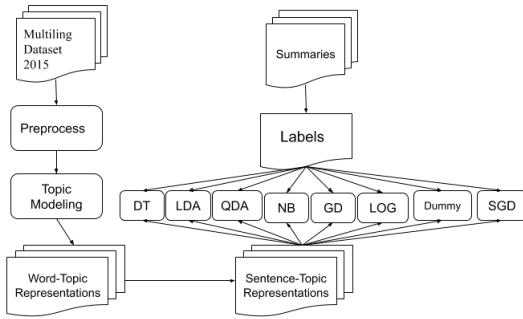


Figure 3.2: Pipeline for topic-modeling based extractive summarization.

3.2.3.1 Binary classification modelling

Extractive summarization can be modelled as a binary classification problem, where one class represents the sentences to be included in the summary, and the other one the sentences that should be ignored. More formally, a document comprised of N sentences $S = \{s_i\}, i \in \{1, \dots, N\}$ is transformed to a subset of M sentence summaries $O = \{o_j\}, o_j \in S, j \in \{1, \dots, M\}$ via a classifier that maps each sentence to a binary label (denoted inclusion in the summary or not). The classification model should select O such that the concatenation of its sentences should produce a coherent, non-reductive and readable summary.

In this study, we tackle classification as a supervised learning procedure; it is necessary to have a set of ground truth sentences, that is, sentences that are indeed valid summaries of input documents. Such data (commonly referred to as “golden” summaries) are manually compiled by humans, who are considered the best summarizers [195]; if a human reader can not differentiate between a human summarizer and an automatic summarizer, that means that the extractive model is optimal. Using the input documents and ground truth data, the classification system can facilitate learning using input sentence features towards a saliency detection model that implements sentence selection towards extractive summarization. We detail this process in the next sections.

3.2.3.2 Topic-based Sentence Extraction

In our approach of extractive summarization, we utilize the topics’ information in word-level feature vector representations using an LDA-based topic model with Gibbs sampling.

The intuition behind our proposed method follows two statements:(1) the significance of a word is reflected by its contribution to a set of semantic topics (2) the significance of a sentence is reflected by patterns in its words-topics contributions.

For the purpose of formality we provide the mathematical description of the proposed

method.

Given a finite set of semantic topics $T = \{T_1, T_2, \dots, T_{|T|}\}$ over the documents' space D , a set of sentences per document $S_{D_i} = \{s_1, s_2, \dots, s_k\}$, and a set of words per sentence $W_{S_{D_i}} = \{w_1, w_2, \dots, w_n\}$, we define the word-topics contribution function of a word w as:

$$C(w) = [p(w, T_1), p(w, T_2), \dots, p(w, T_{|T|})] \quad (3.1)$$

where the vector $C(w)$ is the contribution of the word w to the topics set T and $p(w, T_i)$ is the probability of w being generated by the topic $T_i \in T$ (after the topic model has inferred the posterior probability distributions), as defined by LDA's term-topic distribution. In simpler terms, this probability is computed using:

$$p(w, T_i) = \frac{N(w, T_i)}{N(T_i)} \quad (3.2)$$

where $N(w, T_i)$ and $N(T_i)$ are the number of occurrences of w in T_i and the total number of word occurrences in T_i , respectively.

Further, normalization is applied over the contributions of each word vector, in order to project the values into the $\{0, 1\}$ interval, dividing each value by the maximum value in each vector :

$$C_i(w) = \frac{C_i(w)}{\max(C(w))} \forall i \in \{1, |T|\} \quad (3.3)$$

where $\max(C(w)) \neq 0$

After all word-topic contributions have been calculated, each sentence $s = \{w_1, w_2, \dots, w_n\}$ is represented by the vector

$$C(s) = [C(w_1), C(w_2), \dots, C(w_n)], \quad (3.4)$$

effectively transforming an input set of sentences $S = \{s_1, s_2, \dots, s_k\}$ into the multi-dimensional vector

$$S' = [C(s_1), C(s_2), \dots, C(s_k)] \quad (3.5)$$

Since most machine learning algorithms work with data of equal dimensionality, we apply padding to enforce a uniform dimension across sentences. In zero-padding, the smaller-sized vectors are appended with zeros until all vectors have the same number of dimensions. Since there can be sentences with different dimensions in the documents examined, we implement zero-padding, in order for the elements of S' in equation (3.5) to become uniform.

	train	test
mean num. sentences	233	184.9
mean summ. sentences	77.9	13.5
mean num. words	25.5	22.8
sample sentences	6990	5546

Table 3.4: Multiling 2015 single-document summarization dataset characteristics.

3.2.4 Experiments

3.2.4.1 Dataset and Preprocessing

We use the Multiling 2015 dataset for single-document summarization [207]⁸. The dataset is constructed by the MultiLing community [122] from Wikipedia pages, using articles annotated by human-curated summaries. It consists of 40 languages, spanning 30 documents and summary sets – in our work, we restrict the evaluation to the English language, i.e. work with the 30 English documents provided.

We modify the dataset in order to align it with the extractive summarization setting (as the provided summaries are not purely document sentences). First, the ground truth is modified, labelling input source sentences with a label $l \in \{0, 1\}$ (1 if the sentence should be included in the summary, else 0). This is computed by measuring the similarity of each source sentence with each human-authored summary for the document, in terms of common n-grams. I.e., each human-authored sentence g_i is assigned to a maximally similar source sentence s_j . Stopword filtering is applied prior to this process, and each source sentence is assigned to at most one ground truth sentence.

Additionally, since the dataset used contains very unbalanced classes – the grand majority (with a ratio approximately 13 to 1) belonging to class 0, i.e. the class for sentences that should not be included in the summaries. To alleviate this, we employ an oversampling scheme. To limit the bias towards class 0 during the training phase of our model, we implemented oversampling, by repeating the sentences belonging to class 1 a fixed number of times arriving at a 2 : 1 negative to positive ratio, at most. This way, a classifier that always predicts a dominant label (in this case 0) has sub-optimal performance.

Also, all letters were converted to lower-case in order for the model not to differentiate between words in the beginning and in the middle of sentences, such as "apples" and "Apples". In addition, stop words were also removed from the vocabulary to limit its size, without significant loss of information.

Other preprocessing tasks such as stemming were also explored; however, they did not have a significant effect on the classification performance. After these steps, we end up with the final version of the dataset which is described in detail in table 3.4.

⁸<http://multiling.iit.demokritos.gr/pages/view/1516/multiling-2015>

Metric	DT	KNN	GB	NB	LDA	QDA	Dummy	LOG	SGD
macro-f1	0,497	0,511	0,514	0,514	0,527	0,080	0,452	0,527	0,481
micro-f1	0,898	0,900	0,918	0,911	0,903	0,083	0,643	0,883	0,927

Table 3.5: TF-IDF sentence classification results.

3.2.4.2 Evaluation

We use the provided training and test dataset portion to train and evaluate the produced classifiers. The evaluation is performed in terms of micro and macro F-measure; the former is calculated by counting the total true positives, false negatives and false positives while the macro-averaged variant calculates metrics for each label, and finds their unweighted mean (i.e., not considering label imbalance). Additionally, we compare the predicted summaries with the ground-truth as described in section 3.2.4.1, using the Rouge metric to assess performance [373]⁹. Rouge scores reflect the overlap of n-grams between the ground-truth and the predicted summaries.

3.2.4.3 TF-IDF Sentence Classification

As a baseline model, we also implemented a TF-IDF representation of the input dataset. The TF-IDF scores for each word-document pair are calculated, and each sentence is represented by the vector of the TF-IDF values of the words it contains. For example, a sentence with N words results in a N_w -dimensional vector, where N_w is the number of words in the sentence.

The pipeline for sentence classification using the TF-IDF approach is summarized schematically in Figure 3.1. The scikit-learn v0.21.3 machine learning library¹⁰ is used for building and training the models.

3.2.4.4 Topic Modeling-based Classification of sentences

For the production of the topics and the topic-vectors we used MALLET, a Java framework for various common tasks in NLP, including topic-modeling [422]. Using this tool, we inferred topics over the corpus of the documents in the training set. We subsequently represent firstly the words, and lastly the sentences, of the documents in the training set by their topic-contributions as described in section 3.2.3.2. By default, MALLET ignores all 1-letter and 2-letter words. Additionally, we use the NLTK English stop-words list for stop-word filtering¹¹.

We test the trained topic model by extracting word and sentence-level probabilistic vector

⁹<https://pypi.org/project/py-rouge/>

¹⁰<https://scikit-learn.org/stable/index.html>

¹¹<https://gist.github.com/sebleier/554280>

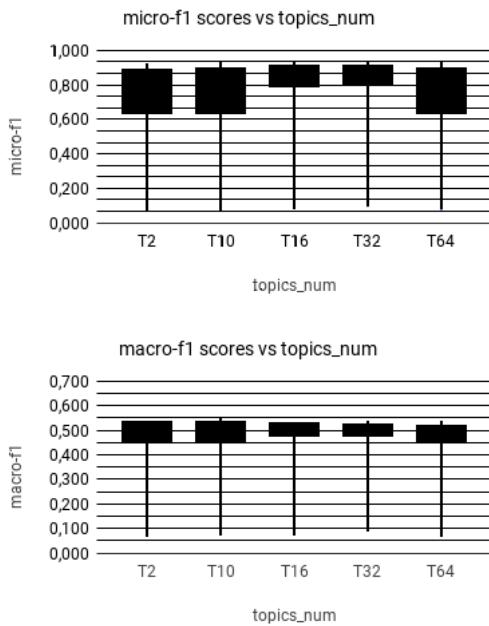


Table 3.6: Comparison of the micro (top) and macro (bottom) f1 performance of topic modeling, based on the selection of the number of topics.

representations from the test set. Any word in the test set not present in the training set, is represented as a zero-vector of topic-contributions.

The pipeline for sentence classification using the topics-based approach can be visualized in figure 3.2 and is outlined below:

- Infer k topics using MALLET's topic model from the training set
- Represent each sentence in the training set using the equation (3.4).
- Train a classifier on the topics-represented training set
- Represent each sentence in the test set using the trained model from Step 1
- Predict the labels in the represented test set
- Evaluate the classifier using the micro and macro f-measures

3.2.5 Results and Discussion

3.2.5.1 Classification Results

The experimental results of the classification on the Multiling Dataset, evaluated with the micro-f1 and macro-f1 scores are displayed in tables 3.5 and 3.7, for the TF-IDF representation and the topic-based representation, respectively. Baseline results using a simple rule-based classifier (Dummy) are also reported, generating predictions with respect to the training set's class distribution – it is thus not influenced by the representation. Dummy gives a micro-f1 score of 0.643 and a macro-f1 score of 0.452.

MICRO F-MEASURE										
Topics	DT	KNN	GB	NB	LDA	QDA	Dummy	LOG	SGD	
2	0,885	0,812	0,884	0,922	0,879	0,102	0,643	0,754	0,073	
10	0,889	0,752	0,895	0,078	0,873	0,175	0,643	0,782	0,927	
16	0,889	0,802	0,908	0,080	0,884	0,928	0,643	0,800	0,927	
32	0,894	0,864	0,909	0,093	0,866	0,927	0,643	0,813	0,927	
64	0,892	0,872	0,916	0,149	0,861	0,927	0,643	0,856	0,073	
mean	0,890	0,820	0,902	0,264	0,873	0,612	0,643	0,801	0,585	
std	0,003	0,044	0,011	0,330	0,008	0,387	0,000	0,034	0,418	
MACRO F-MEASURE										
Topics	DT	KNN	GB	NB	LDA	QDA	Dummy	LOG	SGD	
2	0,535	0,524	0,535	0,489	0,531	0,101	0,452	0,513	0,068	
10	0,532	0,506	0,546	0,073	0,535	0,172	0,452	0,520	0,481	
16	0,528	0,513	0,534	0,076	0,533	0,505	0,452	0,517	0,481	
32	0,537	0,517	0,531	0,091	0,522	0,481	0,452	0,521	0,481	
64	0,517	0,512	0,516	0,149	0,539	0,484	0,452	0,527	0,068	
mean	0,530	0,514	0,532	0,176	0,532	0,349	0,452	0,520	0,316	
std	0,007	0,006	0,010	0,159	0,006	0,175	0,000	0,005	0,202	

Table 3.7: Topic modeling results in micro and macro F1 score.

For TF-IDF, the best macro-f1 score recorded is 0.527 achieved by the Linear Discriminant Analysis (LDA) and Logistic Regression Classifiers (LOG) and the best micro-f1 score is 0.927, given by the Stochastic Gradient Descent Classifier (SGD). TF-IDF achieves significantly better classification results than Dummy , improving micro-f1 by 28% and macro-f1 by 7%, verifying the effectiveness of simple bag-of-word approaches.

For the topic-based representation of sentences, we ran the topic model with different numbers of topics k on each run, and we trained various classifiers for the task. One major limitation of topic-modeling is that the number of topics needs to be determined experimentally. In order to make an informed decision on k , we examined candidate values for the number of topics, visualized in box-plots presented in table 3.6. By analyzing table 3.7 and the box-plots, we concluded that a satisfactory number of topics is 10 for this particular task, as for this k , the Gradient Boosting Classifier (GB) records the highest macro-f1 score. Our decisions are biased towards the macro-f1 instead of the micro-f1 score, since even after the over-sampling of the dataset, the classes are still heavily imbalanced. In addition, we are mostly interested in the sentences that should be included in the summary, which belong to the smaller class. One thing to note, is that as the topic dimension increases, the macro-f1 performance of the Quadratic Discriminant Analysis classifier increases rapidly between Topics 2 and Topics 32 where it reaches a plateau at macro F1 ≈ 0.48 .

Topic-modeling improves on the measures of TF-IDF and Dummy, with a 0.928 micro-f1 score given by the Quadratic Discriminant Analysis (Topics 16) and a 0.546 macro-

f1 score given by Gradient Boosting Classifier(Topics 10) resulting in a 3.6% increase in performance, in comparison with the TF-IDF macro-f1 score. The worst-performing classifiers for the selected number of topics are the Naive Bayes (NB) and Quadratic Discriminant Analysis classifiers.

Finally, considering across-topics averages, SGD, QDA and NB appear to be the least stable configurations, while GB, LDA and DT are among the top performers.

3.2.5.2 Rouge scores

The rouge scores of the summaries produced by the representation-classifier combinations are displayed in tables 3.8 and 3.9. Even though we observed considerable differences in the classification phase between the two representations overall, the final rouge scores are more similar than expected. Bold values correspond to the maximum f-measures for each rouge-metric.

For the TF-IDF, the highest rouge-scores across all classifiers were given by the Quadratic Discriminant Analysis (QDA), while for the Topics-representation, the highest values were recorded by the Naive Bayes Classifier (NB) and Gradient Boosting (GB). The TF-IDF representation results in slightly better rouge-1 to rouge-4 scores while the Topics-based representation produces better rouge-l and rouge-w scores.

3.2.6 Conclusions

To summarize, in this investigation we examined the contribution of topic-based sentence classification to extractive summarization. We examined a variety of configurations for topic modeling by examining a wide range of topics, along with a set of different, diverse classification algorithms. A subsequent large-scale evaluation was performed using micro-f1 and macro-f1 scores. Based on the trained models, we produced summaries for the input documents and we compared them with the ground-truth using several Rouge-metrics. As a baseline, we also implemented a TF-IDF representation of sentences, which follows a traditional bag-of-words weighted approach.

Initial results of this early study show that topic-modeling can be beneficial for sentence classification, as it outperforms the TF-IDF representation, as illustrated by the micro and macro f1 scores in our experiments, albeit this not being the case for the Rouge-based evaluation. We demonstrated that the topics-based approach can easily compete with the TF-IDF approach and shows promise in extractive summarization. Careful task-specific adjustments need to be made however, as the results in the summary evaluation (using Rouge) appear underwhelming compared to those in the classification phase.

Extensions of the study could include more sophisticated methods such as Principal Component Analysis(PCA) [292] or Linear Semantic Analysis(LSA) [339] can be applied on the presented framework of topics-based sentence representation, in order to project the word-topic vectors into lower-dimensional spaces.

CLASSIFIER				METRIC			
		rouge-1	rouge-2	rouge-3	rouge-4	rouge-l	rouge-w
KNN	recall	0,226	0,042	0,013	0,007	0,170	0,034
	precision	0,307	0,056	0,017	0,008	0,232	0,127
	f1	0,245	0,046	0,014	0,007	0,186	0,051
LDA	recall	0,127	0,025	0,008	0,003	0,096	0,019
	precision	0,161	0,036	0,017	0,011	0,120	0,065
	f1	0,136	0,027	0,010	0,004	0,103	0,029
GB	recall	0,164	0,032	0,008	0,004	0,132	0,026
	precision	0,258	0,060	0,019	0,013	0,199	0,113
	f1	0,186	0,038	0,010	0,005	0,149	0,040
LOG	recall	0,153	0,031	0,009	0,003	0,115	0,023
	precision	0,184	0,038	0,012	0,006	0,136	0,071
	f1	0,162	0,034	0,010	0,004	0,122	0,034
QDA	recall	0,365	0,106	0,047	0,026	0,264	0,056
	precision	0,364	0,106	0,047	0,027	0,264	0,140
	f1	0,365	0,106	0,047	0,027	0,264	0,080
Dummy	recall	0,344	0,076	0,029	0,014	0,242	0,050
	precision	0,345	0,076	0,028	0,014	0,243	0,125
	f1	0,344	0,076	0,029	0,014	0,242	0,071
NB	recall	0,208	0,034	0,006	0,001	0,148	0,029
	precision	0,232	0,037	0,006	0,001	0,164	0,082
	f1	0,216	0,035	0,006	0,001	0,154	0,042
DT	recall	0,280	0,043	0,010	0,003	0,207	0,041
	precision	0,323	0,045	0,010	0,003	0,239	0,122
	f1	0,292	0,044	0,010	0,003	0,216	0,060
SGD	recall	0,000	0,000	0,000	0,000	0,000	0,000
	precision	0,000	0,000	0,000	0,000	0,000	0,000
	f1	0,000	0,000	0,000	0,000	0,000	0,000

Table 3.8: TF-IDF Rouge Scores

CLASSIFIER				METRIC			
		rouge-1	rouge-2		rouge-3	rouge-4	rouge-l
KNN	recall	0.326	0.06	0.019	0.01	0.238	0.048
	precision	0.332	0.062	0.019	0.01	0.241	0.122
	f1	0.328	0.061	0.019	0.01	0.239	0.069
LDA	recall	0.365	0.105	0.046	0.026	0.268	0.057
	precision	0.332	0.063	0.018	0.009	0.236	0.118
	f1	0.327	0.062	0.018	0.009	0.232	0.066
GB	recall	0.334	0.069	0.025	0.015	0.242	0.049
	precision	0.361	0.105	0.046	0.026	0.265	0.14
	f1	0.361	0.104	0.046	0.026	0.265	0.08
LOG	recall	0.362	0.102	0.045	0.025	0.267	0.056
	precision	0.339	0.069	0.025	0.015	0.245	0.125
	f1	0.336	0.069	0.025	0.015	0.243	0.071
QDA	recall	0.305	0.064	0.022	0.012	0.221	0.045
	precision	0.362	0.101	0.045	0.025	0.267	0.141
	f1	0.361	0.101	0.045	0.025	0.267	0.08
Dummy	recall	0.344	0.076	0.029	0.014	0.242	0.05
	precision	0.345	0.076	0.028	0.014	0.243	0.125
	f1	0.344	0.076	0.029	0.014	0.242	0.071
NB	recall	0.313	0.063	0.021	0.013	0.228	0.046
	precision	0.364	0.104	0.046	0.026	0.268	0.142
	f1	0.364	0.104	0.045	0.026	0.267	0.081
DT	recall	0.363	0.105	0.046	0.025	0.265	0.056
	precision	0.33	0.065	0.022	0.013	0.241	0.125
	f1	0.319	0.064	0.022	0.013	0.232	0.067
SGD	recall	0.323	0.062	0.018	0.009	0.23	0.046
	precision	0.331	0.067	0.022	0.012	0.238	0.124
	f1	0.312	0.065	0.022	0.012	0.226	0.066

Table 3.9: Topic modeling Rouge Scores

Additionally, more adaptive topic modelling approaches could be applied, removing the need for pre-determined topic specification [597]. Moreover, Neural Network classification architectures can be explored, in addition to the set of classifiers we already tested on the dataset. A-priori knowledge on words, phrases and sentences from external sources (e.g. knowledge bases such as Wordnet [1]) could also prove beneficial for the training phase of the machine-learning models. Finally, future work will take order / target summary length into account, making our results comparable to other systems tackling the Multiling2015 dataset and the state of the art.

3.3 Automatic Summarization of Video Game Reviews

In this work we continue our examination of the summarization task by performing an initial investigation on automatic summarization applied on video game reviews, which is a domain both rich and often neglected by summarization studies.

3.3.1 Introduction

The ever-expanding popularity of digital games is evidenced by the large profit margins of the commercial game industry sector [42], the vast and diverse swathes of the population that play games [163], and the appeal of games and gamification beyond the purposes of entertainment [147]. A large factor for the market penetration of digital games are distribution platforms such as *Steam* and the *Google Play Store*. Not only do these distribution platforms allow interested players to purchase and download new games, they also cultivate a player community with players returning to rate and comment on their favorite game or even contribute user-created content, strategies, cheats, etc. This community-driven content often informs other users' purchases (e.g. via an aggregated review score) but is also carefully monitored by developers and publishers in order to gauge opinions on specific aspects of the game which can be patched or improved in updates to the game or in sequels. For both players and developers, being able to succinctly monitor other players' views is highly beneficial. The website www.metacritic.com aggregates reviews by players and professional critics, returning a percentage score for the game and highlighting diverse reviews along the spectrum of positive versus negative. The Steam platform also aggregates its users' reviews into different categories ('Mixed', 'Overwhelmingly Positive', 'Mostly Negative' etc.) which is another criterion for sorting and (likely) promoting games. The simple aggregation of reviews into a general score is important, but it obfuscates the nuances of the different reviewers' grievances and is of limited use to designers who wish to improve their game. This research work explores techniques for text summarization in order to provide a multi-dimensional and holistic summary of Steam reviews for a particular game.

We explore the topic of summarization for game reviews using a large dataset of Steam reviews from 12 selected games. The goal of the summarization pipeline is to extract users' views on different facets of games such as graphics, audio, and gameplay [367],

leveraging textual sentiment analysis to identify positive and negative review snippets, creating a composite summary of indicative comments on a specific game facet. Unlike the numerical aggregation of Metacritic or Steam, this approach extracts individual sentences (and criticisms) contained within a usually dense review and attempts to classify those in terms of positive or negative automatically (rather than based on the user’s binary recommendation). The presentation of the game’s summary, which is split based on different aspects typically criticized in games, can be valuable for both players and designers. For players, the statistics derived from this process (e.g. ratio of positive versus negative comments in one aspect) can act as an expanded game scoring system not unlike professional game reviews which gave a score to graphics, audio etc. For designers, the indicative comments split per sentiment and aspect allows for a quick monitoring of players’ current favorite features. Moreover, the flexible way in which aspects are defined allows designers to explicitly redefine the keywords they are interested in, personalizing the summary to their design priorities.

There has been very limited attention to game review summarization, besides student projects [722]. Inspired by the only work that performs aspect-based game review summarization [687], this study evaluates the outcomes of a straightforward summarization pipeline in a small-scale user survey. Using the twelve most reviewed games in a 2017 dataset of Steam reviews, the resulting summaries are evaluated by a small set of experts. The work studies pipeline variants to better sketch what is important in game review summarization. Based on the outcomes of the different summarization processes, and a small-scale study where the different outcomes were compared, a number of potential improvements were identified. The study also highlights the many directions which game review summarization research can follow so that it can serve designers and players through different pipeline implementations, alternative visualizations, bottom-up aspect discovery, or text processing driven by domain knowledge.

The rest of this investigation is structured as follows. We start with a review of related works in Section 3.3.2. We then describe the proposed summarization pipeline and variants in Section 3.3.3. We describe the dataset in Section 3.3.4 and present two different user studies in Sections 3.3.5 and 3.3.6. We then discuss the results in Section 3.3.7 and conclude the study in Section 3.3.8.

3.3.2 Related Work

User reviews are a rich source of information, although the extraction and analysis of this information can be challenging not only due to the textual nature of the medium but also because users tend to have a mixed opinion about various features [452]. Approaches such as sentiment analysis as well as summarization have been applied to various datasets, such as product reviews [452, 258], movie reviews [720, 715], or hotel reviews [260]. Section 3.3.2.1 surveys relevant approaches for the different phases of a summarization pipeline, while Section 3.3.2.2 discusses the nuances of the Steam platform and early work in game review summarization. For interested readers, [260] provides a more thorough overview on review summarization according to the type of corpora used as input.

3.3.2.1 Summarization Pipeline

Summarization can be *extractive* when relevant portions (usually sentences) of the input are copied and combined, or *abstractive* when new text is generated to rephrase and summarize the input [171]. The summarization pipeline requires a number of steps before the raw textual input can produce a summary; algorithms and approaches for each step are discussed below.

A fundamental step towards summarization (and natural language processing more broadly) is the pre-processing and extraction of features from the dataset. In the analysis below, the term “documents” is used to describe any type of text, e.g. a sentence, a paragraph, or an academic paper. One popular if naive approach for pre-processing data is the *bag-of-words* which collects all words in the document, disregarding their order and grammar. This method counts the number of instances of the same word, and the frequency of occurrence of each word is used as a feature to measure similarity between documents. Since many words (such as articles or pronouns) are far more frequent in all documents, terms are weighted based on their frequency via $tf.idf$ [546] where the term frequency (tf) is multiplied by the inverse document frequency (idf). Unlike the bag-of-words approach, the word order is considered in many other approaches as it can capture a word’s importance. For instance, the first and last sentences in a larger document tend to be more important [469]. Other approaches tag words on their part-of-speech (POS) [508], e.g. nouns (NN), verbs (VB), or adverbs (RB). This is useful for pre-processing, e.g. selecting only sentences with a noun and adjective as a corpus for review summarization [260]. Another use of POS tags is to select N-grams (i.e. a sequence of words) with specific parts of speech, such as a comparative adverb followed by an adjective [626].

Identifying the topic of a document, sentence, or review is often necessary for clustering opinions on the topic together. When the topics of interest are known in advance, experts usually provide the keywords used to filter the relevant documents. For instance, TweetElect used an initial set of 38 keywords related to the 2016 US elections (including candidates’ names) for streaming relevant tweets [136]. However, a boolean check whether a keyword is specifically mentioned is rarely sufficient due to the nuances of language; *query expansion* is applied to create a larger set of terms related to each original keyword [405]. Supervised learning is often applied for topic modelling, showing positive and negative examples of relevant documents to a classifier [405]. When topics are unknown and must be discovered from the data, a simple approach is to identify the most frequent terms and cluster emergent terms based on co-occurrence [162]. Probabilistic topic models such as Latent Dirichlet Allocation (LDA) [66] can more efficiently discover topics without domain knowledge, following a bag-of-words approach which disregards word or document order. LDA randomly chooses a set of topics and decomposes the probability distribution matrix of words in a document into two matrices consisting of the distribution of topics in a document and the distribution of words in a topic. Due to the vast number of possible topic structures, sampling-based algorithms are used to find the sample topics which best approximate the posterior distribution [65]. LDA has often been applied to find topics within reviews, primarily in order to identify review’s sentiments to-

wards these topics, e.g. in [269].

The sentiment behind utterances is important for summarization, especially when the corpus is reviews of any kind. Turney [626] highlighted that reviews may recommend or not a certain product, movie, or travel destination; a summary therefore should account for both positive and negative reviews. Turney’s study was the first to perform sentiment analysis on text-based reviews based on responses of the AltaVista internet search query on how near the phrases were to the word ‘excellent’ (for recommended) and the word ‘poor’ (for not recommended). Manually created lexicons for words that express sentiment have been used in conjunction with fuzzy logic, vector distance, etc. to classify positive and negative [137, 601]. In the same context, there has been extensive work on extracting opinion words which express subjective opinions within sentences [662]. It has been found that subjective sentences are statistically correlated with the presence of adjectives [662], and much research in product review summarization uses adjectives to determine sentiment polarity. For instance, Hu *et al.* [258] used a frequency-based algorithm to find relevant domain features, and then extracted nearby adjectives to such domain features. Using a labeled set of adjectives and expanding the initial set via WordNet, Hu *et al.* classified the extracted adjectives’ polarity and assigned that positive or negative sentiment to the nearby domain feature. The SentiWordNet database is constructed based on the same principles of the domain-specific adjective classification of [258], using a manually annotated set of seed words and using WordNet term relationships to expand the training set, which is then used as the ground truth for machine learning classifiers [29]. SentiWordNet, and similar general-purpose models for sentiment prediction [616], have been used for polarity detection in reviews, e.g. in [540, 233].

3.3.2.2 Steam Review Summarization

Since its 2003 release, the Steam platform has become the largest digital distribution platform for PC gaming [157], hosting over 34,000 games and tens of millions of active users daily. This study focuses on user-created reviews on Steam, although other initiatives such as the Steam workshop allow users to upload their mods or strategies and comment on others’ content. User reviews can be submitted only by people that have purchased the game from Steam, although they are visible to all. As noted above, Steam aggregates user reviews into a category and provides a number of companion statistics, including a timeline of reviewer’s scores. Reviews themselves consist of a single binary recommendation (Recommended versus Not Recommended) and a text explaining the user’s opinion. Other users can review the quality of the review itself by tagging it helpful, not helpful, funny, or breaking the Rules of Conduct. By default, Steam shows the most helpful reviews submitted within the last 30 days, although users can also choose to sort reviews by other criteria.

As noted in the introduction, there is no systematic academic research in Steam review summarization. To the best of our knowledge, the only academic publication that tackles the problem of aspect-based summarization on such data is by Yauris and Khodra [687]. In their approach, only relevant portions of sentences were extracted via conditions

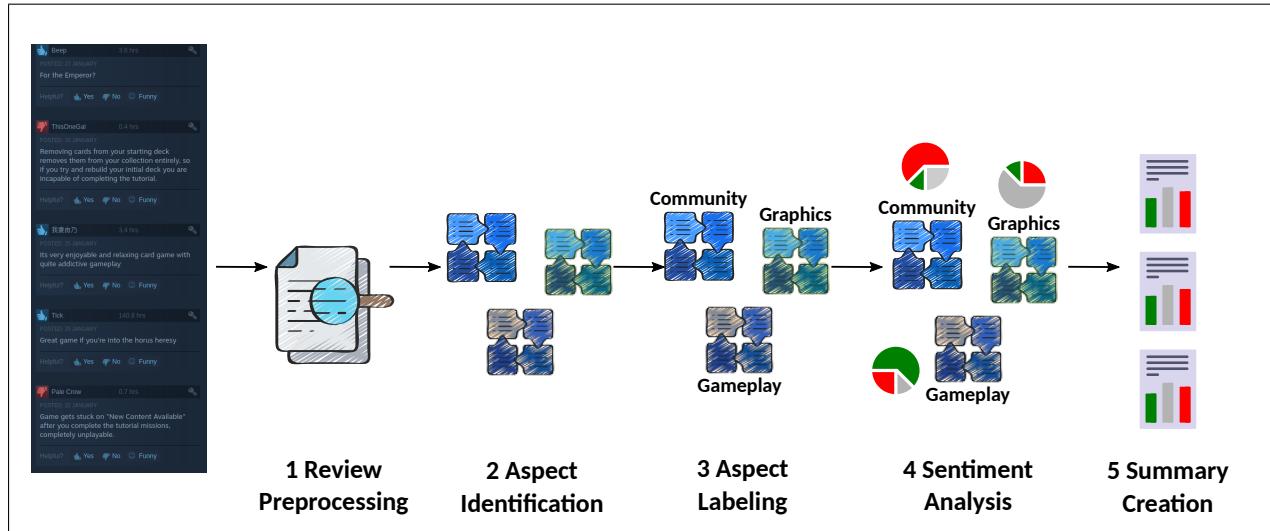


Figure 3.3: The full pipeline represents both the Clustering variant (CL Full) and the Deep Learning variant (DL Full), while variant CL AsDe produces summaries by skipping the Sentiment Analysis step.

applied on text tagged via Parts of Speech; these portions were usually small, e.g. the phrase could be “amount of content” [687]. Similar to our approach, a pre-specified set of keywords are used for aspect categorization. The aspects and keywords are similar but not identical to our approach (e.g. the aspects in [687] are gameplay, story, graphic, music, community, and general/others), while choosing the aspect described in the phrase was based on the cosine similarity from each word of the phrase to the aspect’s keywords. The output summary consists of many aspects (most of which are outside the pre-specified keywords) and a single adjective for each, unlike our current work which extracts complete sentences with different polarities. The summarization pipeline was tested on a single game (*Skyrim*), exploring different sentiment extraction approaches using precision and recall as performance metrics. While our current work does not explore as many parameters for sentiment analysis, it is the first instance where game review summaries are evaluated by humans in a small-scale but thorough user study.

3.3.3 Summarization Pipelines

Figure 3.3 visualizes the main components of our pipeline:

Preprocessing which aims to prepare the input reviews for further analysis. This may imply cleaning, chunking text in snippets or sentences, Part-of-Speech tagging, and other similar tasks.

Aspect Identification which identifies interesting aspects (or topics) in the reviews. These topics may be expressed as a set of words, e.g. “visual, aesthetic, scenery” or “soundscape, audio experience, sound effects”.

Aspect Labeling which assigns clear, descriptive labels to the discovered aspects. E.g. "graphics", "audio".

Sentiment Analysis which gathers information related to the sentiment expressed within the reviews. This information may later be used to update the final summary appropriately. For example, one may need only positive views in the summary, or—most probably—a sampling of all the views, be they positive or negative.

Summary Creation which implies the process which, given all the information gathered in previous steps, forms and renders the final summary for the user.

Given the above pipeline, we implemented three different variants. The first two are based on keyword detection and *Clustering* (CL). The first variant does not do Sentiment Analysis, while the second one uses the full pipeline. The last one is another full pipe method based on *Deep Learning* (DL) that focuses on improving on Aspect Labeling and Summary Creation steps.

3.3.3.1 CL pipeline

During the preprocessing step, each review is split into sentences, each sentence is cleaned in order to create the basic elements on which the final summaries will be based. The cleaning process included some character replacements so that each sentence could be presentable (e.g. starting with a capital letter and ending with a period) even if it originated from a larger sentence that was split during sentence splitting. Moreover, preprocessing prepared the lemmatized versions of the sentences which are used for aspect detection. In these lemmatized sentences, general stop words are removed. For all preprocessing steps, we used the default functions (and stop word lists) of the *nltk* Python library [63].

The aspect detection process is split into two parts: aspect identification and aspect labeling. Aspect identification splits sentences into sets that focus on a specific aspect while aspect labeling identifies this aspect in order to present it to the final review summary.

Our approach uses a predefined set of aspects, presented in Table 3.10. We selected these six aspects since they are well-established facets of games [367] and are popular dimensions within professional reviews.

A simple approach for aspect labeling is to use a dictionary of keywords per aspect as the ones presented in Table 3.10. In order to be able to include sentences even when they do not include the exact keywords, a k-means clustering is applied to all sentences to find clusters with similar text. Terms are weighted based on their frequency via TF-IDF, which has been used extensively for sentence similarity in bag-of-words approaches (see Section 3.3.2.1). The result is K clusters of sentences with similar words to each other; in all our experiments we set $K = 20$ based on prior evidence [480]. Once sentences are all assigned a cluster based on the distance to the center, all sentences in all clusters are processed in the following fashion:

Aspect	Keywords
Graphics	graphic, visual, aesthetic, animation, scenery
Gameplay	mission, item, map, weapon, mode, multiplayer, control
Audio	audio, sound, music, soundtrack, melody, voice
Community	community, toxic, friendly
Performance	server, bug, connection, lag, latency, ping, crash, glitch, optimization
Story	dialog, romance, ending, cutscene, story

Table 3.10: Aspects and keywords used for the identification of dominant aspects in review clusters.

1. If the sentence contains the exact keywords of only one aspect, the sentence is assigned to that aspect and is flagged as a candidate that can be used by the summary of that aspect.
2. If keywords from multiple aspects are found in the sentence, the sentence is flagged as an unsuitable candidate for any summary and removed.
3. If no aspect keywords are found in the sentence, the most common aspect within the sentences of the same cluster will be used to label this sentence and flag it as a candidate. For instance, if a sentence does not contain any keyword, but sentences in its cluster predominantly belong to the aspect *Gameplay* via case (1), then the sentence is also assigned to the same aspect and flagged as a candidate.

Using the sentences from cases (1) and (3), a set of candidate sentences is created per aspect. Using these sets, the first variation of our pipeline could now produce a summary. This variation, named Clustering Aspect Detection summary (CL AsDe), chooses N sentences at random from each aspect's set. A sample CL AsDe summary can be found in Table 3.11 for Tom Clancy's The Division.

The next step of the process is Sentiment Analysis, which is used by the next summarization variant (CL Full). Using the different sets of candidate sentences per aspect, the sentiment polarity (positive or negative) of each sentence is calculated by averaging the sentiment score of each word it contains. As above, sentiment analysis of each word is done via the default functions of the *nltk* Python library [63]. The library calculates probabilities for each polarity class (positive, neutral, negative). We took into account sentences which were assigned a class with a probability of at least 0.5. In order to select a number of sentences per category, a Kmeans clustering approach (using TF-IDF) is applied within the set of sentences with the same polarity. In the CL Full implementation of this study,

only two sentences per polarity are selected ($k = 2$) as the ones closest to each cluster's centroid. If there exist sufficient positive and negative sentences, then this approach returns 6 sentences as bullet points. Note that if fewer than two sentences are above the threshold for positive (or below the threshold, for negative) then fewer sentences may be included in the summary. An example summary from CL Full variant can be found in Table 3.11 for Tom Clancy's The Division.

3.3.3.2 DL pipeline

After experimenting with the first two variant pipelines and taking into account the feedback of the first user study (see Section 3.3.5), we decided to focus on improving the following:

- Keyword detection and clustering based Aspect Labeling must be improved to avoid sentences such as "If those things all sound good to you you will like the game." to be labeled as audio sentences.
- The final summary should somehow provide information regarding the whole sentiment of the given aspect and not just by the selected sentences.
- The final summary should use a better sentence extraction approach in order to deal with redundancy.

Taking all the above into account, the DL pipeline makes changes to the Aspect Detection and Summary Creation steps of the CL pipeline described in Section 3.3.3.1.

For Aspect Detection, we used the BERT model [148] to generate embeddings for game reviews. BERT is a deep neural language model that uses a bidirectional, multilayer transformer architecture, exploiting cross and self-attention to capture word interdependencies effectively [31, 639]. The approach relies on multi-head attention modules for sequence encoding modelling, with word order information being retained with additive positional encoding vectors. BERT is trained in an unsupervised setting on large quantities of English text, using masked language modelling and next sentence prediction objectives. These tasks require the prediction of hidden sequence tokens and the generation of an entire sequence, given an input sequence (e.g. for tasks such as question-answering and text entailment, etc.). This pretraining scheme and architecture have been shown to perform exceptionally well for a variety of natural language understanding tasks.

To obtain the representation for a game review, we feed the text to the model using a sequence length of 16 tokens. We use the $BERT_{BASE}$ model variant, that produces 768-dimensional sequence embeddings, learned during training for classification purposes. The implementation and pre-trained model utilized are provided by the transformers software package from *huggingface*¹². Using the produced embeddings as features we trained a binary Ridge Logistic Regression classifier [172] (one vs all) for each aspect. We also trained a seventh classifier to detect sentences unfit for any aspect. For each candidate

¹²<https://huggingface.co/>

sentence a confidence score was calculated by each aspect classifier. Only sentences with a high prediction confidence in the given aspect and a low confidence on each other classifier were selected as summary candidates for the next steps of the pipeline.

During the Summary Creation we applied the following strategy to the 100 most probable candidate sentences of each aspect. First, the NewSum Toolkit [206] was used to select the sentences that provide the most representative information. NewSum uses language-agnostic methods based on n-gram graphs, that not only extract the most representative sentences, but also deal with redundancy. In the end we had 20 candidate sentences per Aspect. The final summary was composed by 6 sentences using the following strategy:

- Select the most positive sentence (Sentiment Analysis).
- Select the most negative sentence (Sentiment Analysis).
- Select the first 3 sentences provided by the NewSum Toolkit (excluding the previously selected sentences).
- Create an artificial sentence using the polarities provided by Sentiment Analysis of all the aspect sentences. The polarity of each sentence was mapped as 1, 0 or -1 (positive, neutral, negative) using thresholds. Given an Aspect and the mean Polarity score \bar{P} , the possible produced sentences reflect opinions that fall in the following categories:
 - **Mixed:** $\bar{P} \approx 0$, high standard deviation.
 - **Mostly neutral:** $\bar{P} \approx 0$, low standard deviation.
 - **Mostly positive:** $\bar{P} > 0$ above a threshold.
 - **Mostly negative:** $\bar{P} < 0$ below a threshold.

The final summary is composed by randomly shuffling these 6 sentences. An example summary from DL Full variant can be found in Table 3.11 for Tom Clancy's The Division.

3.3.4 Dataset

As a first demonstration of the summarization pipeline, we follow [480] and select the most helpful reviews on Steam, splitting them per game. This study parses the Steam review dataset gathered by Zuo [722], which consists of over 7 million reviews obtained via Steam's API. Each review text comes with a plethora of features concerning both the game being reviewed and the reviewer, although only a subset of features is used for this experiment. Since Steam users can vote a review as helpful, unhelpful, or spam, we only consider 'valid' reviews those with 10 or more user votes as 'helpful'. With this criterion (minimum of at least 1000 of 'helpful' reviews), we select twelve games with the most valid reviews (see Table 3.12). The games selected have a desirable diversity both in terms of genres (shooting, survival, adventure, open-world, multi-player, single-player, etc.) and in terms of general audience reception (shown by the Metacritic score which aggregates professional and users' reviews).

<ul style="list-style-type: none"> - In a few words the game is single dimensional this might sound vague but it becomes apparent that there is not much depth as you play once you're a couple hours in. - Clothes sound "right" when you move in them. - They sound good and looked good with ability to mod for better stats or even rerolling stats. - They have improved the pve portion of the game and crazy as it sounds the pvp too. - No music and something feels so strangely abandoned about it. - Like how if there's a blizzard your cap and shoulder will be covered in snow and that npc voices will echo when they are standing in hallways with hollow walls. - Very good voice acting. - Great abilities pretty good sounds; indoor echos reverb off objects etc. - If those things all sound good to you you will like the game. - Superb voice acting and ambient city sounds are also a good plus for this game. - It sounds hyperbolic but I'm being dead serious. - Sounds terrible right
<ul style="list-style-type: none"> - Most opinions are positive regarding audio. - The voice acting in the game is in the higher tiers as is most ubisoft games. - There are not a lot of different voices and some of the voice acting for them is bad. - Ubisoft - bugs - the textures are so fucked up that nobody can play this game anymore. - And it clearly shows I want to play it and that I try to. -I'm gonna be honest the cinematics are pretty great.
<ul style="list-style-type: none"> -

Table 3.11: Summaries generated by different pipelines, for aspect *Audio* of Tom Clancy's The Division. From top to bottom: CL AsDe (only aspect detection), CL Full (aspect detection with sentiment analysis) and DL Full (Deep learning combined with a sophisticated summarizer).

Game Title	Publisher	Year	Reviews	MC
No Man's Sky	Hello Games	2016	4146	61%
DayZ	Bohemia Interactive	2018	3349	–
PAYDAY 2	Starbreeze	2017	2573	79%
ARK: Survival Evolved	Studio Wildcard	2017	2368	70%
Grand Theft Auto V	Rockstar Games	2015	2104	96%
Firewatch	Campo Santo	2016	1599	81%
Darkest Dungeon	Red Hook Studios	2016	1564	84%
Just Survive	Daybreak Game Company	2015	1463	–
Killing Floor 2	Tripwire Interactive	2016	1276	75%
Elite Dangerous	Frontier Developments	2015	1270	80%
Tom Clancy's 'The Division'	Ubisoft	2016	1091	79%
Subnautica	Unknown Worlds Entertainment	2018	1056	87%

Table 3.12: Games selected from the dataset, sorted by the number of ‘valid’ reviews (10 or more ‘helpful’ votes). The Metacritic score (MC) is included for reference.

For each of the selected games we selected to keep the 10 thousand most up-voted reviews. As already discussed in Section 3.3.3 each of these reviews was split into sentences to create a sentence pool per game. On average, the sentence pool consisted of around 50 thousand sentences per game. The smallest pool of sentences was for PAYDAY 2 (37K), while the largest one was for Elite Dangerous (70K). The average length of the sentences was 85.7 in characters and 16.4 in words. In terms of both characters and words, the longest sentences were those of Darkest Dungeon (average of 91.8 characters and 17.3 words) and the shortest ones were those of Just Survive (average of 79.9 characters and 15.6 words).

In terms of aspects, the most common one was *Gameplay* on average. *Performance* was the next most popular aspect and in certain games such as ARK: Survival Evolved it was the most popular one. The least popular aspect was *Audio* with a ratio of 1 to 5 compared to the *Gameplay* aspect.

In terms of sentiment, the majority of sentences were more neutral than positive or nega-

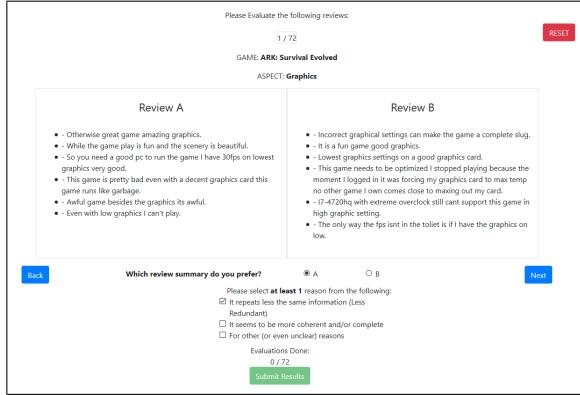


Figure 3.4: User interface for online evaluation of summaries produced by CL AsDe and CL Full methods.

tive. Between positive and negative sentiment, no general safe conclusions can be drawn since the results varied given different combinations of aspects and games. In general, we can say that the aspect *Performance* was characterized as negative more frequently. The opposite was true for the aspect *Graphics*. On the other hand the sentiment ratio (positive vs negative) towards the aspect *Community* varied between different games.

3.3.5 First User Study

As a first experiment, we evaluated the two variations of the CL pipeline (CL AsDe and CL Full) in a small-scale user-study with summaries of aspects of the 12 games of Table 3.12.

3.3.5.1 Annotation Protocol

A pairwise comparison process was followed, rather than a scale-based rating approach, due to (a) evidence that comparison-based evaluation can be less demanding cognitively [115] and (b) a rich body of literature that has applied pairwise evaluation for summarization tasks [474] (e.g. the single document summarization task in [207]).

To this end, we created an online evaluation user interface (UI) (see Figure 3.4) which supported comparative pairwise evaluation of summaries. We initialized the system by providing two sets of summaries \mathbb{A}, \mathbb{B} , one from system A and one from system B . Each summary in \mathbb{A} corresponded to a summary in \mathbb{B} , as they both summarize the same set of reviews and the same aspect (e.g. the aspect *Graphics* of DayZ). During the experiment, each system's summary was randomly placed first or second to minimize any bias related ordering effect.

The UI also informed the user of the title of the game being summarized, plus the aspect (e.g. *Graphics*). The user was then called to select their preferred summary (A or B) and

explain the reasons for this preference. For the latter annotation, the user could select one or more tickboxes among the following options:

- It repeats less the same information (Less Redundant)
- It seems to be more coherent and/or complete
- For other (or even unclear) reasons

The first two options aim to assess whether redundancy is a concern and, similarly, whether coherence and completeness are useful in the task. Redundancy has been traditionally a summarization evaluation indicator [14], especially in multi-document summarization. The completeness and coherence aspect is essentially a (more nuanced) version of overall responsiveness, as this has been used in DUC/TAC summarization tracks and related work [133].

3.3.5.2 Participants

The evaluation was carried out by eight adult evaluators (3 female), fluent in English, with gaming experience. The evaluators were selected explicitly among the authors' network of contacts and invited directly by the authors. Participants were asked to connect to the online system and evaluate all 72 pairs of summaries (produced by CL AsDe and CL Full), which covered all predefined aspects (see Table 3.10) of the ten games in Table 3.12. There was no time limit for completing the evaluation, but there was a requirement that all pairs were evaluated in a single session.

3.3.5.3 Results

The data collected from the experiment was a total of 576 observations, including the preference of each evaluator for each pair of summaries and the reasons for this choice. The primary goals of the user study are to assess (a) whether the annotators prefer one of the two summarization approaches, (b) which criteria they explicitly (via the three tickboxes) or implicitly (based on properties of the summary) consider when selecting their preference. Towards this end, the data is processed based on the 8 users' annotations on 72 game/aspect pairs (for a total of 576 data points), and all statistical tests are performed at a 5% significance threshold. Our assumption is that the complete CL pipeline which includes both aspect detection and sentiment analysis will offer a richer and more diverse summary than AsDe alone.

Regarding users' preference of one summarization technique, results were mixed: overall, annotators had no clear preference with CL AsDe being marginally more often selected (53%). Table 3.13 shows the distribution of selection of CL Full split per aspect. The Table shows that the main factor for the skew of the overall preference towards CL AsDe was the graphics summaries, as the other aspects are fairly evenly preferred between the two approaches.

Aspect	CL Full	CL AsDe
Audio	43%	57%
Community	51%	49%
Gameplay	55%	45%
Graphics	30%	70%
Performance	54%	46%
Story	49%	51%
Overall	47%	53%

Table 3.13: First user study: annotators' preference of one summarization algorithm over the other, per aspect and overall.

	Df	F value	p value
game	11	1.519	0.120
aspect	5	3.912	0.001 *
evaluator	6	7.945	0.000 *
coherence	1	18.6491	0.000 *
redundancy	1	5.7604	0.017 *
other	1	0.5639	0.453

Table 3.14: Analysis of variance between the preference of one approach and different factors. Significant findings are shown with an asterisk. The analysis is made on the F statistic and the degrees of freedom (Df) are also noted.

To further assess which factors led to the annotators' preference of one summary over the other, we conducted an analysis of variance test (ANOVA) between the preferred approach (represented as a binary choice) and other features such as the aspect. Table 3.14 shows the results in terms of significant differences, and verifies that there is a systematic influence between the aspect and preference. On the other hand, the game does not seem to affect users' preference of one summary or the other; this is a promising finding as the methods are supposed to be applicable to any game. There is also clear evidence that preference was highly varying from annotator to annotator, and annotators rarely agreed with each other even in this simple pair-wise preference task.

To get a better understanding of the reasons annotators gave regarding their preference, we looked further into the statistics of the winning observations of CL AsDe vs. CL Full. When AsDe was preferred, annotators explained their preference mainly due to better coherence (63%), lower redundancy (28%), but also 'other reasons' (26%). When CL Full was preferred, annotators chose 'other reasons' (50%), and less often coherence (41%) or low redundancy (17%). This finding shows that summaries by AsDe were more coherent, but annotators still preferred summaries by CL Full often for other reasons. This points to a limitation of the experimental protocol, as the interface did not provide annotators with enough options to allow them to explain their reasons for their summary preference. This was addressed in the second user study (see Section 3.3.6) with an extra option on the UI. It should be noted that better coherence was selected far more

often overall (53% of instances) than lower redundancy (23%), while ‘other reasons’ were also chosen often (37%). Redundancy and coherence were chosen together in only 5% of instances, and thus it is evident that these two axes of evaluation are fairly independent. These findings, coupled with the statistically significant influence (via ANOVA) between preference of summarization approach and tagged coherence and redundancy, support our conclusion that both coherence and redundancy were important factors for annotators’ preference.

3.3.6 Second User Study

Based on the findings and limitations identified in the first user study, conducted a second study with more participants but fewer games, testing the best CL approaches with the novel DL Full pipeline. Due to participants’ concerns on the long duration of the 72-item survey in the first experiment, we opted to use only two games to lower the time required from annotators; it is expected that fatigue would likely introduce noise to the participants’ responses. Details on how the games and annotation options were chosen are detailed in Section 3.3.6.1.

3.3.6.1 Annotation Protocol

The user interface for the second user study was largely the same as in the first (see Section 3.3.5.1). Based on the first study’s finding that ‘other reasons’ for an annotator’s preference were often chosen, a fourth option was added to the UI as a tickbox stating “The summary was more focused and contained less irrelevant information.” We refer to this additional option as Focus in the analysis that follows.

As noted above, to reduce the time required for the study only two games were chosen to be annotated. We chose among the games from the first user study, taking the game where CL Full had the highest preference (Tom Clancy’s The Division, where CL Full was chosen 60% of the time) and the game where CL AsDe had the highest preference (Elite Dangerous, where CL AsDe was chosen 60% of the time). For each of the two games, the preferred method was chosen to present to the user, juxtaposed with the summary for the same game and aspect produced by DL Full. Therefore, the participant had to annotate 12 items, 6 aspects for Tom Clancy’s the Division comparing the CL Full summary with the DL Full summary and 6 aspects for Elite Dangerous comparing the CL AsDe summary with the DL Full summary. The rationale was to select the most successful game summaries (for both CL variants) and compare them with the novel DL pipeline. We refer to CL and DL summaries in this study, referring to the best CL summary (CL Full or CL AsDe) as shown to the user.

As with the first user study, the order of the two options was randomized (i.e. sometimes CL summaries were shown first, sometimes second). Unlike the previous experiment, however, the order of the sentences within the same summary was also randomized; the

rationale was to avoid ordering effects when the participant starts by reading an incoherent sentence first.

3.3.6.2 Participants

Fourteen participants completed this annotation task. Unlike the previous study, a snowball method for soliciting participants was followed, soliciting feedback from a broader group. Thus, this study lacks data on the demographics and gaming experience of participants, although participants were all adults and had experience in data analysis and artificial intelligence.

3.3.6.3 Results

The data collected from the experiment was a total of 168 observations. Overall CL summaries were slightly more preferred by participants (55%), although the difference is not statistically significant (Paired t-test, p-value 0.22). Interestingly, for Elite Dangerous (which was summarized by CL AsDe) the difference was more pronounced (CL AsDe preferred 60% of the time over DL Full); for Tom Clancy's The Division the two methods (CL Full and DL Full) were chosen evenly. Since only one game was tested per CL variant, it is difficult to assess whether the preference was due to the game itself or the sentiment-based selection component. Moreover, while DL Full includes sentiment-based selection, this part accounts for 2 of the 6 sentences and thus it is even more difficult to estimate the reasons for the users' preference. This ambiguity points to further refinements needed for the annotation protocol which is discussed in Section 3.3.7.

In terms of the reasons offered by participants for their choice, coherence was still most commonly chosen (62% of responses), followed closely by focus (56%). Low redundancy was chosen less often (23%), while 'other reasons' are chosen only in 14% of responses. The addition of the focus option seems to have mitigated the prevalence of 'other reasons' in the first study. Unlike the first study, however, low redundancy was often chosen in conjunction with one other reason (56% of the time) or two other reasons (36% of the time). Combined with its low overall prevalence, it is possible that low redundancy may now longer be necessary as a separate reason in the UI, although a broader user study with more games is needed to validate this hypothesis.

Pearson's Chi-squared tests were also used in order to test whether any of the above reasons is correlated to the preferred summary. Only redundancy was found to be correlated with the type of summary (p-value 0.001). This clearly indicates the importance of handling redundancy satisfactorily in any future approach.

3.3.7 Discussion

This work introduced a number of possible pipelines for identifying, grouping, and extracting the opinions of users in terms of pre-specified game facets. Two small-scale user surveys examined the preference of users in the presence of different pipeline implementations. Results indicate that (a) aspect extraction is important for summarization, although deep-learning does not necessarily improve the aspect extraction process compared to a simpler clustering-based method; (b) between the clustering-based pipeline variants (CL AsDe, CL Full), there was no clear winner with respect to the summary outputs; (c) evaluators had strong and individual opinions on which variant was better; (d) sentiment-based criteria and/or confidence-based criteria for selecting sentences do not seem to perform better than the random selection performed by CL AsDe.

While the aspects chosen for this experiment were intuitive, based on typical facets of games that players and professional critics focus on, some resulting aspect-based summaries were less coherent than others. The choice to assign a sentence to an aspect even if its cluster only had a slim majority in keyword frequency likely introduced inconsistency. For CL aspect detection, the most significant factor for the lack of coherence was the choice of keywords. Specifically, the keyword “sound” was often found in sentences unrelated to game audio, used as a verb: e.g. “On paper this game sounds great”. To a degree, such artefacts were removed in the DL aspect detection pipeline via (a) the latent sentence representation and (b) fine-tuning the model based on manual annotations on this specific corpus. However, a more sophisticated method for aspect detection seems necessary. For instance, an adaptive query expansion as followed by [405] could create a much larger set of keywords automatically, although it may overlook the nuances of game terminology. On the other hand, a Word2Vec model [437] trained on the entire corpus of steam reviews (or even larger game-related corpora such as game FAQs and fansites) could be used to derive a similarity score with specific aspects. Building a game ontology for this task or using an existing one [488, 538] could further assist in discovering more keywords or in calculating an ontology-based semantic similarity measure [548]. Finally, a completely different direction could see the discovery of topics specific to each game rather than focusing on the same pre-specified topics every time. This would be valuable as different genres have a different focus (e.g. multiplayer games focus on balance or lag, while horror games focus on the emotional response), but could make it difficult to maintain the same presentation format across games and thus confuse end-users.

Sentiment analysis was also often problematic, primarily due to the informal and idiosyncratic language that games reviews were often in. Reviews are often rife with sarcasm and negation, e.g. “Have fun spending huge amounts of hours for very little progress.”. Moreover, many reviews’ sentences have poor syntax and are very short or very long (e.g. “Good: + great aesthetic.”). Sentiment analysis treated the sentence as a bag-of-words, exacerbating the problem. In general, sentiment analysis can not capture negation or sarcasm and handles incomplete sentences poorly. Performance would likely be improved with a more appropriate pre-trained lexicon for informal utterances on the Social Web, such as SentiStrength [616] or other sentiment- and negation-aware approaches [213].

Alternatively, a custom classifier for sentiment analysis could be trained using text from a Steam review as input and the user’s recommendation as polarity. Complementing the training set with experts’ annotations could refine such a model, especially when dealing with sarcasm. Another promising alternative to SentiWordNet for sentiment analysis would be the use of an authored dictionary of opinion words [258] or game-specific adjectives annotated in terms of polarity [697].

Our findings also showed no clear winner between the two CL variants or between CL and DL summaries. This ambiguity of the findings could well be by-products of the experimental protocol followed. Findings from the first user study pointed to a missing reason for players to report, and the second study included a “focus” reason which improved the quality of the data collected but raised questions about the importance of the “low redundancy” reason. The users’ reported fatigue in the first experiment led to fewer items in the second study to alleviate the burden from annotators. However, this increased the locality of the findings in the second study as it was unclear whether preferences were due to the game or the algorithm. In future studies, summaries for more games should be annotated by more participants, showing only two games to each user but randomizing which games are shown when the user starts the study. More importantly, the current experimental protocol forces participants to select one review as preferred and provide at least one reason. The forced choice between two summaries does not allow the user to provide more nuanced feedback. A four-alternative forced-choice (4-AFC) with options “A”, “B”, “both A and B”, “neither A nor B” would allow the user to point out cases where both summaries are equally good or equally bad. The fairly even split between the two alternatives in both user studies could be due to the fact that users consider some summaries shown equally bad and select randomly. On the other hand, a 4-AFC questionnaire would likely need many more participants since much of the data will be removed when no ranking is given. The need for more games, more annotation choices, and perhaps more algorithm variants (DL AsDe, for instance) point to the need for a large-scale user survey among the general gaming community, which will be performed in future work in this vein.

As discussed in Section 3.3.1 and explored on a high-level during the user study, game review summarization can be valuable both to consumers (players) and producers (game developers). However, each stakeholder has different priorities and will likely respond differently to different summary formats. The extractive summarization process was visualized as ‘pure text’ bullet points, which was not as engaging to either type of audience. It would be important to explore alternative visualizations for players and developers. For players, the summary could provide more structure (based on pre-specified game facets), focus more on the weights and scoring of each aspect (including visualizations such as pie-charts), show only a few polar opposites in terms of review sentences, and perhaps cross-reference these findings with other games’ review summaries. For developers, on the other hand, a bottom-up topic discovery would likely be beneficial in order to identify unexpected points of contention among users. Moreover, presenting the context of the reviewers’ chosen sentences would also be valuable for designers, e.g. how many reviewers agree with or echo this comment, when this comment was made and whether general sentiment has shifted since then. Such context can be important regarding the

urgency of addressing certain concerns or to gauge whether patches and updates have improved reviewers' perception, not unlike Steam's use of most recent reviews.

There are many directions for future research depending on the purpose of the game review summarization. As a tool for game evaluation, primarily targeted towards players or producers, the game's context is important in order to choose which reviews or topics to highlight. Additional research in this vein would need to find topics or patterns in similar games (e.g. of the same genre, publisher, or publication date) and then to compare the current game's reviews in terms of those topics or compared to other games' reviews. User experience research would also be important to find how best to present such results, as interactive summaries where the user can zoom in and out into different games and/or different topics within games would make the summaries more intuitive and manageable. As a tool for game analysis, bottom-up probabilistic topic modelling [65] in games of the same genre could help identify design patterns [64] and players' expectations based on their repertoire [299]. As a tool for knowledge discovery, game reviews can serve as raw text or multi-modal corpora from which structured data can be automatically extracted as entities and relations [590], concept hierarchies [549, 701], or even a complete game ontology [458, 511].

3.3.8 Conclusion

This study highlighted the challenges and opportunities of game review summarization via natural language processing. We introduced a pipeline for grouping Steam users' comments into pre-specified aspects such as visuals or performance, and studied different renderings of the final summary, exploiting positive and negative sentences based on sentiment analysis. The small-scale user survey revealed differences in how different annotators assess the reviews, highlighted possible foci of research for better game review summarization systems, and suggested a number of refinements to the process are suggested in this promising subfield of game artificial intelligence.

3.4 Clustering, Summarization and Classification of Web Documents and Social Media

This section tackles we tackle multiple machine learning tasks for textual content, namely clustering, summarization and classification from articles and social media posts, integrated in a rich event / change detection and tracking system.

3.4.1 Introduction

In remote sensing, *change detection* is the process of comparing two or more satellite images that depict the same area on the Earth's surface, but are taken at different points in time [396, 514]. Its goal is to identify differences between the images in the form of

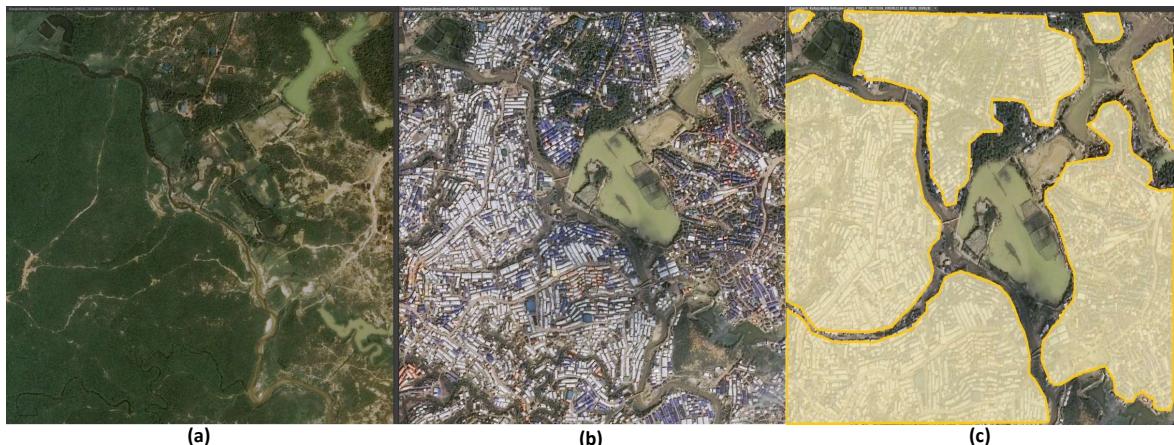


Figure 3.5: Satellite images showing Ukhiya, Chittagong, Bangladesh (a) before, and (b) after the Rohingya refugee crisis in October, 2017. (c) shows the main areas with changes in land cover or land use as identified by GeoSensor.

areas with changes in land cover or land use (e.g., an area that was an olive grove in the past is now occupied by buildings). This is a crucial task, as it provides useful information for many applications, e.g., studying land cover evolution, monitoring natural disasters or support to crisis management. As an example, consider Figures 3.5(a) and (b), which depict snapshots of Ukhiya, Chittagong, Bangladesh before and after the settlement of Rohingya refugees in October, 2017. In situations like this, change detection allows for fast and accurate estimation of natural or man-made changes on the Earth’s surface, providing valuable support to decision-makers. In our example, the outcomes of change detection appear in Figure 3.5(c). Modern satellite technology makes this possible even for remote areas with humanitarian or security issues that are difficult to reach.

Interest in change detection using satellite images has grown recently, due to the availability of long time series of images by flagship Earth observation programmes, such as the US Landsat program¹³ and the EU *Copernicus Programme*¹⁴. The latter is currently the world’s largest Earth observation programme with almost 20 satellites, called *Sentinel*s, expected to be in orbit by 2030. It consists of a set of complex systems that collect data from satellites as well as in-situ sensors, providing reliable and up-to-date information on a range of environmental and security issues under a free, full and open data policy. Information extracted from this data is also made freely available to users through the *Copernicus services*¹⁵, which address six thematic areas: land, marine, atmosphere, climate, emergency and security. Techniques for change detection using time series of satellite images are important in all of these areas [74].

To the best of our knowledge, though, there is no open-source system that addresses the following three Vs of Big Satellite Data:

¹³<https://landsat.usgs.gov>

¹⁴<http://www.copernicus.eu>

¹⁵<http://www.copernicus.eu/main/services>

- *Volume* stems from the combined effect of the inherently quadratic time complexity of change detection and the large size of satellite images. In the worst case, all pixels of the one image have to be compared with all pixels of the other image, yielding a rather time-consuming procedure for a common pair of images - each image typically occupies few GBs, containing millions of pixels of low resolution (i.e., each pixel corresponds to tens of square meters on the Earth's surface). Apparently, change detection poses a quite challenging computational task for commodity hardware.
- *Veracity* requires that decision makers are able to assess the quality and correctness of the intelligence extracted from satellite images, based on relevant news content. In practice, this means that *collateral information* about news should provide reliable insights into the detected changes, ideally in real-time.
- *Variety* emanates from the diverse types of images that are produced by each satellite constellation. The two polar-orbiting satellites of the Sentinel-1 constellation are equipped with C-band Synthetic Aperture Radar (SAR) imaging systems, which enable image acquisitions regardless of weather and light conditions (i.e., the sensor is able to acquire images in the presence of clouds and during night time). In contrast, the two polar-orbiting satellites of the Sentinel-2 mission provide High-Resolution Optical data, acquired by a wide swath high-resolution multispectral sensor. Their images have 12 spectral bands, covering the spectrum from the visible domain to the short wavelength infrared domain. Being an optical passive system, imaging is sensitive to weather conditions and depends on external illumination. Variety further increases due to the textual data that are necessary for addressing Veracity.

In this work, we present GeoSensor, a geospatial system that applies change detection to Copernicus data in a way that addresses these three Vs of Big Satellite Data. In essence, GeoSensor integrates a remote sensing component with a social sensing one into a highly scalable processing chain. Remote sensing applies change detection techniques to SAR images from Sentinel-1, while using optical Sentinel-2 images for the validation of the end result. Social sensing applies event detection techniques to cluster together news items and social media posts that pertain to the same real-world event and are located in the area where change detection took place. For example, Figure 3.6(a) depicts a cluster of news items that elucidates the changes appearing in Figure 3.5(c). The integration of these two orthogonal components relies on Semantic Web technologies.

The rest of this section is structured as follows: Section 3.4.2 briefly discusses related work, while Section 3.4.3 delves into GeoSensor's architecture, highlighting the three workflows that lie at its core. In Section 3.4.4, we present preliminary experiments over real-world data to demonstrate the scalability of our system and in Section 3.4.7, we conclude the study along with directions for future work.

3.4.2 Related Work

Change Detection. *Earth observation* is the use of remote sensing technologies to monitor land, marine and atmosphere. Satellite-based Earth observation relies on the use of

satellite-mounted payloads to gather imaging data about Earth characteristics. We can distinguish two kinds of remote sensing. *(i)* In *passive remote sensing*, the satellite instruments monitor the energy received from the Earth, due to the reflection and re-emission of the Sun’s energy by the Earth’s surface or atmosphere. Optical or thermal sensors are commonly-used passive sensors (e.g., Sentinel-2 images). *(ii)* In *active remote sensing*, the satellite sends energy to Earth and monitors the energy received back from the Earth’s surface or atmosphere, enabling day and night monitoring during all weather conditions. Commonly used active sensors are lasers and radar images, like the SAR images provided by Sentinel-1.

Recent works on change detection use Deep Neural Networks [218, 308] in a data-driven fashion, performing classification to detect changes in pixels or areas in the images. Other works use hierarchical object-based classification methods [100]. Such *supervised algorithms*, though, lie out of our scope, due to the lack of publicly available labeled datasets. Developing such datasets from scratch is a rigorous process that requires heavy human involvement, even in-situ inspection of identified changes.

Instead, GeoSensor considers *unsupervised algorithms* for change detection. At the moment, it is equipped with the established approach implemented in ESA’s SNAP Toolbox¹⁶. Yet, its modular architecture allows for seamlessly extending it with additional state-of-the-art approaches, like the clustering technique in [167].

Event Detection. A review of text event detection is presented in [685], with more recent surveys covering a large variety of detection methods that are crafted for social media [26, 481]. In [83], the authors utilize a semantically-enabled convolutional neural network (CNN) to categorize social media posts, reporting that their model outperforms TF-IDF and Word2Vec pre-trained embeddings. Other works incorporate CNNs for the joint detection of events and topics [83, 104, 465]. Yet, these methods rely on supervised learning, requiring a labeled dataset, unlike our unsupervised approach.

On another line of research, several works use unsupervised, semantically-aware clustering for event detection. For example, a semantically rich multiple-vector representation is used in [406, 407], while [475] uses a co-occurrence-based semantic expansion of words to produce event groups. These works report superior performance over non-semantic baselines. In [550], the authors employ a classification-based cleaning phase that is followed by content- and temporal-based clustering. [3] performs a clustering on keyword-based features over tweets, while the structure of the underlying social network lies at the core of the approaches presented in [11, 329]. However, all these works mainly rely on vector space features that capture frequency-related statistics, ignoring the positional information of tokens in the source text (i.e., bag of n-grams). In contrast, our approach relies on graphs of n-grams, which effectively capture token context both in long, curated documents like news articles and in short, noisy texts like tweets [12, 208, 482].

¹⁶<http://step.esa.int/main/toolboxes/snap>

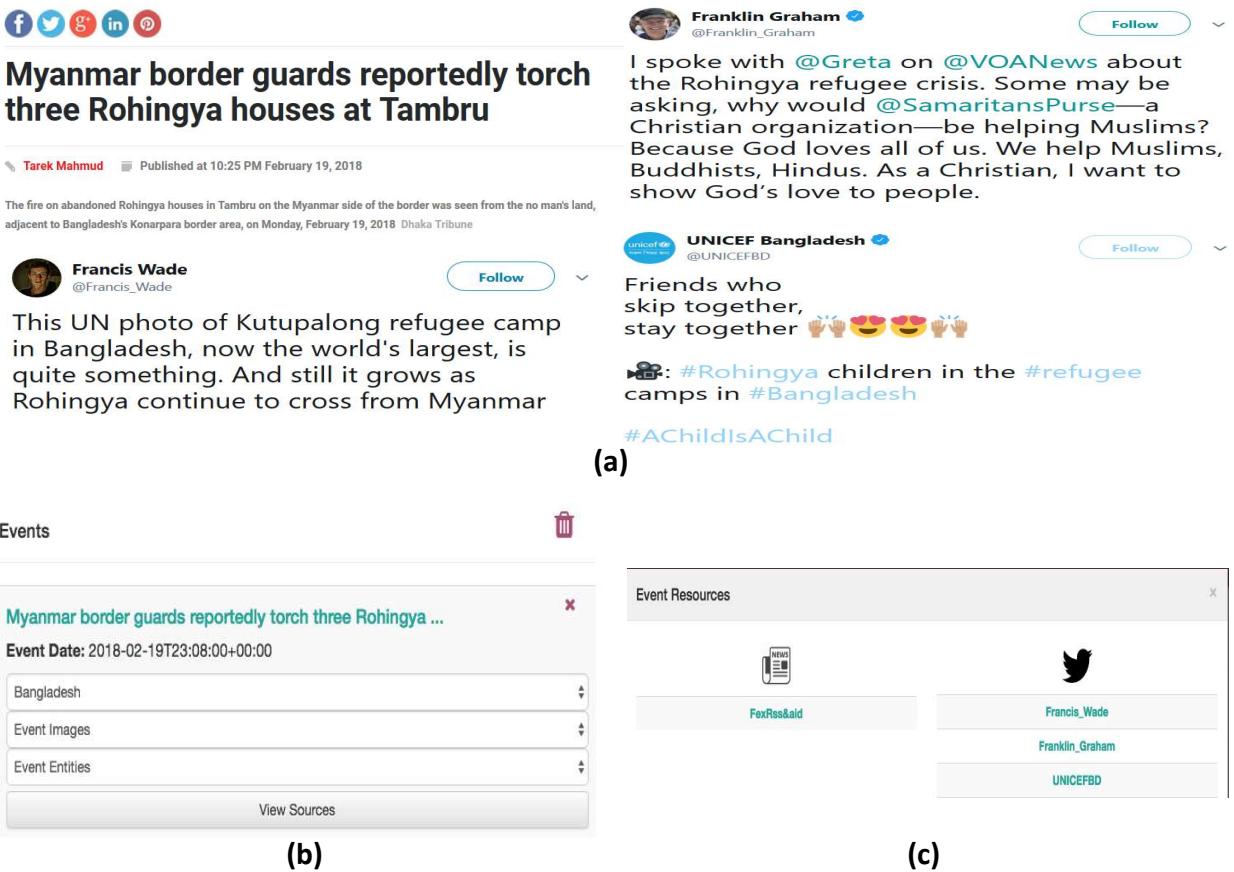


Figure 3.6: (a) A set of news items referring to the Rohingya refugee crises, (b) the corresponding event created by GeoSensor, and (c) the menu providing access to the individual news items of the event.

3.4.3 Approach

We now present GeoSensor, explaining how it addresses the above three Vs of Big Satellite Data.

To tackle Variety, GeoSensor relies heavily on state-of-the-art Semantic Web technologies, which provide time efficient, unified access to the outcomes of the remote and the social sensing components. In this way, it is capable of seamlessly processing a rich diversity of data sources, which range from the graphic information in SAR and optical satellite images to the textual information of news articles and social media posts.

To address Volume, GeoSensor exploits the distributed processing of a cluster based on the *BDI platform* [27], the open-source, semantics-enabled Big Data infrastructure that was developed in the context of the EU BigDataEurope¹⁷ project. The BDI platform combines the massive parallelization capabilities offered by Apache Spark¹⁸ with an inherent

¹⁷<https://www.big-data-europe.eu>

¹⁸<https://spark.apache.org>

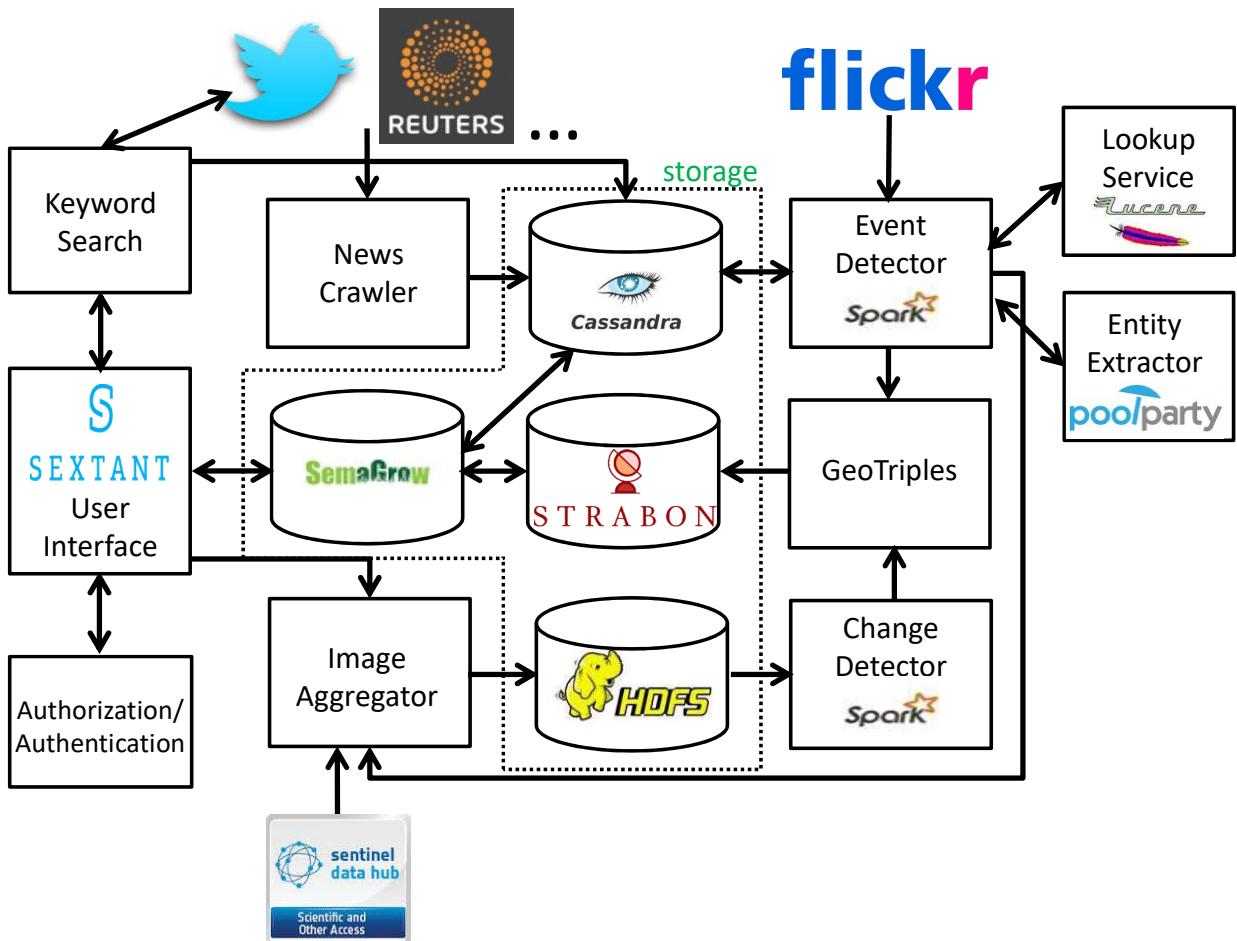


Figure 3.7: The system architecture of GeoSensor.

support for Semantic Web technologies.

To tackle Veracity, GeoSensor uses Sentinel-2 images in combination with the knowledge extracted from the social sensing component for the verification of changes detected from Sentinel-1 images. This is enhanced by the ability to fetch the latest social media data through the live Twitter keyword search offered by its GUI.

Figure 3.7 depicts GeoSensor's architecture. It consists of 11 components that are organized into 3 workflows, one for each horizontal layer: the *change detection layer* is formed by the components at the bottom (i.e., Image Aggregator, HDFS and Change Detector), while the *event detection layer* is implemented by the components at the top (i.e., News Crawler, Apache Cassandra, Event Detector, Lookup Service and Entity Extractor). The rest of the components comprise the *semantic layer*, which acts as GeoSensor's backbone. Next, we describe the functionality of each layer in detail.

3.4.3.1 Change Detection Layer

This layer implements the gist of GeoSensor, retrieving and comparing pairs of satellite images in order to detect changes in land cover or land use. It consists of three components.

The first one is the **Image Aggregator**, a RESTful web service that downloads from ESA's Copernicus Open Access Hub¹⁹ the pairs of Sentinel-1 and Sentinel-2 images with the largest overlap with the user-defined area of interest. In our example, the Image Aggregator is responsible for downloading the images in Figure 3.5(a) and (b), after the user specifies Ukhiya, Chittagong, Bangladesh as the area of interest. This process also requires the user to define temporal acquisition criteria, in the form of the images' sensing dates, i.e., the time of interest together with a reference date in the past, before the change took place. In our example, the time of interest - for Figure 3.5(b) - is October 26, 2017, while the reference date - for Figure 3.5(a) - is anything before June, 2017. The downloaded Sentinel images are then stored to the Hadoop Distributed File System (**HDFS**), distributing parts of the images to all cluster nodes, facilitating scalable and fault-tolerant parallel image processing.

Finally, the **Change Detector** applies the workflow depicted in Figure 3.8, which implements in parallel the state-of-the-art *unsupervised* approach offered by ESA's SNAP Toolbox. Its goal is to compare the downloaded images in order to identify the changes in land cover or land use. This workflow consists of three stages: (i) Pre-processing uses *co-registration* [678] to ensure that the selected images have identical dimensions and correspond to the same geolocation. (ii) Main processing compares the individual pixels in the images to assess their difference. (iii) Post-processing clusters together the pixels with high likelihood of changes, forming broader areas with changes in a way that reduces false alarms, i.e., it excludes outliers caused by *noise*, which is either inherent in the satellite images or introduced by inaccuracies of previous steps.

In more detail, we call *master image* the one corresponding to the earliest date - Figure 3.5(a) in our example - and *slave image* the one corresponding to the latest date - Figure 3.5(b). Typically, their dimensions and characteristics are quite different, because they were taken under different settings, such as the angle of the satellite. Therefore, pre-processing (co-registration) is indispensable for aligning the two images in such a way that each pair of corresponding pixels represents the same point on the Earth's surface.

Given that individual satellite images typically cover a very large area on Earth, the **subset operator** crops the original satellite images to the borders of the user-defined area of interest. This operation curtails the running time to a significant extent, restricting the computational cost to the absolutely essential parts of satellite images. Its complexity is very low, requiring no parallelization.

The cropped images are given as input to the **collocate operator**, which resamples the pixels of the slave image into the geographical raster of the master. This operator requires accurate geopositioning information for both images in the form of *ground control points*

¹⁹<https://scihub.copernicus.eu/>

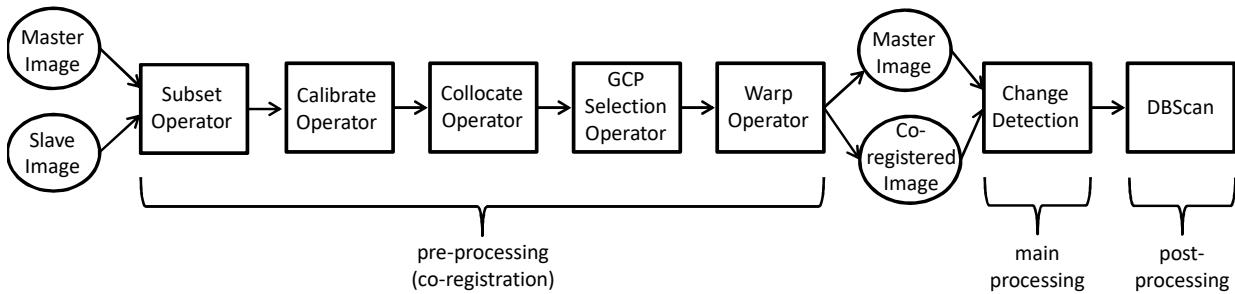


Figure 3.8: The workflow implemented by Change Detector.

(GCPs), i.e., markers for certain geographical positions within a geo-referenced image that are described by their geo-coordinates and by textual descriptions in the image meta-data.

Next, the **GCP selection operator** generates a set of uniformly spaced GCPs in the master image and computes their corresponding GCPs in the slave image. This is done through an iterative process: for each master GCP, the corresponding slave GCP is approximated based on their geo-coordinates. Using a predetermined window size, the areas surrounding each GCP are cross-correlated in order to adjust the slave GCP to a more accurate position. This procedure is repeated until the new slave GCP is located within acceptable limits, or a maximum number of iterations is carried out.

Based on the selected GCPs, the **warp operator** computes the *warp function*, which will be used for mapping the pixels of the slave image into the co-registered image. This is a linear function that is estimated by repeating the following process until convergence: a warp function is initially computed using the available master-slave GCP pairs. The resulting function is used to map the master GCPs to the slave image. Then, the residuals between the mapped master and the corresponding slave GCPs are computed along with the root-mean-square (RMS) and the standard deviation of all residuals. Next, the master-slave GCP pairs are filtered to eliminate those exceeding the mean RMS. Upon completion of this process, the remaining master-slave GCP pairs are filtered with a predetermined RMS threshold and the warp function is derived from the retained pairs.

Finally, the *co-registered image* is generated using the resulting warp function in combination with bilinear interpolation. This means that every point of the original slave image is projected to a point in the master image as the weighted sum of the warp projection of its four surrounding pixels.

Using the master and the co-registered image as input, the **change detection** algorithm computes the ratio of the corresponding pixels in the two images. The pixels exhibiting very large or very low ratios indicate candidate areas with changes.

Lastly, **DBScan** [168] is applied for post-processing the set of candidate areas with changes. DBScan groups together pixels closely packed together (i.e., with many nearby change indicators), while treating as outliers pixels that lie alone in low-density regions, with their nearest neighbors located far away. The end result is a set of areas with changes in land cover or land use. In our example, DBScan produces the image in Figure 3.5(c), yielding the 7 yellow clusters that correspond to such areas.

Due to the high time complexity of all processes (except the Subset operator), they are massively parallelized in Apache Spark. Due to space limitations, we omit the parallelization details.

3.4.3.2 Event Detection Layer

To address Veracity, this layer attaches a set of recent events to every area with identified changes in land cover or land use, providing users with a possible explanation and verification of the detected changes. This functionality is offered by the five components at the top layer of GeoSensor’s architecture in Figure 3.7.

The first component is the **News Crawler**, which scans at half-hour intervals specific social media sources and news agencies for the latest news items (posts and news articles, respectively). For the time being, these sources include most of the RSS feeds that are freely provided by Reuters in English²⁰ as well as several selected public accounts in Twitter²¹, also in English. The crawler structure, though, is extensible, facilitating the integration of more information sources, or even the extension with other operation modes. For example, it has been used as a basic data collection infrastructure in a summarization application [206] and in the EU project “NOMAD”²². In our running example, the News Crawler is responsible for gathering the news items in Figure 3.6(a).

All data gathered by the News Crawler are stored in the second component, namely **Apache Cassandra**²³. We opted for this particular data management system, due to its capacity to store a large volume of information, while offering linear scalability and fault-tolerance (i.e., it provides high availability with no single point of failure). In fact, Cassandra is crafted for large-scale infrastructures like the BDI platform, offering robust support for clusters with multiple commodity servers. Besides, it is an open-source NoSQL database that is compatible with the SemaGrow component, which is used by the semantic layer for federated access to the details of individual news items or entire events (cf. Section 3.4.3.3).

The news items stored in Cassandra are periodically processed by the **Event Detector** module at half-hour intervals. They are grouped into real-world events by a modified version of NewSum²⁴ [206], a summarization algorithm providing commercial-grade performance. NewSum uses n-gram graphs [205] to model its textual input, a representation that has been shown to be effective in noisy settings in multiple genres (i.e., blogs, articles, microblogging and social media) [208, 482]. In addition, NewSum is robust to multilingual data, ranking among the top performers in multilingual, multi-document summarization tasks [207].

In more detail, Event Detector first builds a coarse-grained set of events. Pairs of news

²⁰<https://www.reuters.com/tools/rss>

²¹<https://twitter.com>

²²<http://www.nomad-project.eu>

²³<http://cassandra.apache.org>

²⁴<https://github.com/scify>

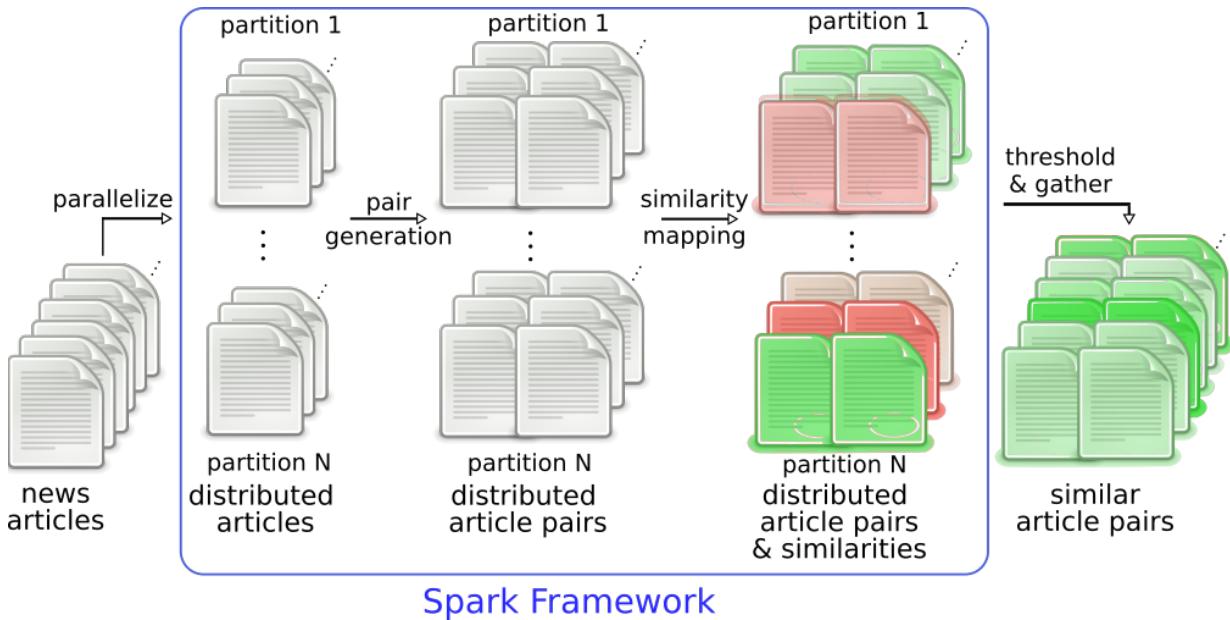


Figure 3.9: The Spark-based implementation of Event Detector.

articles are compared with each other using their n-gram graphs representation and the corresponding graph-based textual similarity measures [205]. Appropriate thresholding is then applied to retain only the pairs with high similarity. Those pairs are then grouped into larger sets (pools) of news articles based on a transitivity analysis that forms clusters from connected components in the similarity graph. The *pools of news articles* with a very low support are discarded, whereas the remaining pools are considered as “real-world events”. Due to its high time complexity, this process is parallelized in Apache Spark, as shown in Figure 3.9. The same procedure is applied independently to Twitter data, yielding a set of *tweet pools*. Each tweet pool is then compared with every pool of news articles. If their similarity exceeds a predefined threshold, the tweet pool is added to the pool of news articles. Then, every pool of news articles goes through a summarization process that builds its event description (e.g., title selection) and enriches it with relevant metadata, i.e., spatiotemporal information, named entities as well as image elements from its member documents. This metadata are extracted from its content directly, or with the help of RESTful-based tools and services, internal (Lookup Service and Entity Extractor) and external ones (PoolParty²⁵ and Flickr²⁶).

In more detail, the **Lookup Service** associates the location names from news items with their actual geo-coordinates so that they can be joined with areas with detected changes in land cover or land use. The location names are identified and extracted from the text data in each news item using Apache openNLP²⁷. In the example of Figure 3.6(a), the location of Kutupalong refugee camp (Ukhiya, Chittagong, Bangladesh) will be converted into the following geo-coordinates: POLYGON ((92.0455551147462 21.3476104736329, 92.2031173706055 21.3476104736329,

²⁵<https://www.poolparty.biz>

²⁶<https://www.flickr.com>

²⁷<https://opennlp.apache.org/>

92.2031173706055 21.1280899047852, 92.0455551147462 21.1280899047852,
92.0455551147462 21.3476104736329)) – note that the output is in the form of the OGC²⁸ standard Well Known Text (WKT).

This conversion may seem a trivial task, given that there is little ambiguity in our example. In reality, though, location names typically suffer from high levels of noise. There are homonymous locations (e.g., London, UK and London, Ontario, Canada) as well as spelling mistakes (e.g., Landon), due to errors in the extraction process. To address both challenges, the Lookup Service poses every place name as a keyword query to an Apache Lucene²⁹ index that contains about 180,000 location names of administrative areas worldwide (GADM dataset³⁰). Lucene's fuzzy query functionality deals with spelling mistakes, while homonymy is addressed by ranking the candidates in decreasing order of the ratio "string similarity/area". The WKT polygon coordinates corresponding to the top ranked location are finally returned as output.

Valuable metadata are also provided by the **Entity Extractor**, which enriches the event description with named entities that are extracted from their textual content, thus empowering a Semantic Web view of the produced information. This view allows for improved indexing and disambiguation of the main players in an event, based on the URIs mapped to each extracted entity. At the core of this functionality lies the **PoolParty Semantic Suite**, which constitutes a state-of-the-art thesaurus management tool that is based on Linked Data [552]. Specifically, a "Famous People" thesaurus was constructed, containing almost half a million entities of well-known actual and fictitious personalities, each grounded to a URI. Two RESTful APIs were implemented and hosted by PoolParty. Given an input text or a news item url, the first endpoint, called Extractor API, provides a list with entities deemed relevant to the supplied content. The entity URIs are stored in Cassandra, along with their corresponding thesaurus id. The second endpoint, called Metadata API, retrieves descriptive metadata related to the entity, whose URI is given as input. These procedures are illustrated in Figure 3.10.

The Entity Extractor also associates every detected event with publicly available images from **Flickr**. Using the Flickr search API, it retrieves photographs geo-tagged within the geolocation(s) of each event that have been uploaded at a close enough date.

Finally, all event descriptions, including their metadata, are stored into Cassandra in the appropriate tables that distinguish them from individual news items. Duplicate events are discarded and Strabon is notified for the new entries (see below for details).

3.4.3.3 Semantic Layer

This layer constitutes GeoSensor's backbone, bridging the gap between the two orthogonal operations of change and event detection. This is achieved by the four components in the middle of Figure 3.7, which encapsulate state-of-the-art Semantic Web technologies.

²⁸The Open Geospatial Consortium - <http://www.opengeospatial.org>

²⁹<https://lucene.apache.org>

³⁰<http://www.gadm.org>

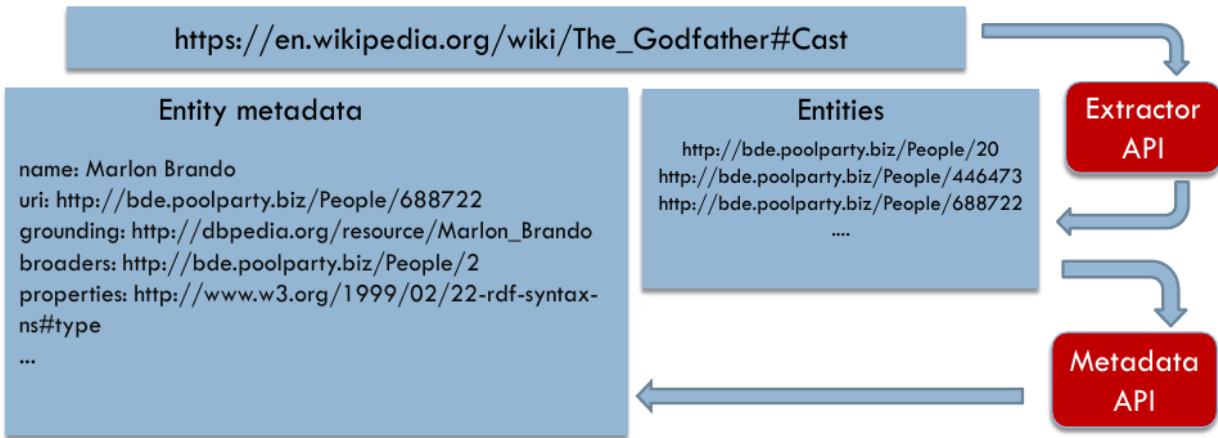


Figure 3.10: Entity extraction example, illustrating the Extractor and Metadata API.

The first component is **Geotriples** [334], a tool for transforming geospatial data from their original formats into RDF. In our case, it converts into RDF the descriptions of areas with changes in land cover or land use (from change detection) as well as the event summaries (from event detection). We selected GeoTriples, as it is an established system that supports a wide variety of data formats [333].

The output of Geotriples is stored into **Strabon** [332], a state-of-the-art open-source spatio-temporal triplestore that efficiently executes GeoSPARQL and stSPARQL queries. Strabon supports spatial datatypes, enabling the serialization of geometric objects in the OGC standards WKT and Geography Markup Language (GML). It has been implemented by extending the established RDF store Sesame (now called RDF4J³¹), using the spatially-enabled database PostGIS³². Strabon is the most efficient spatio-temporal RDF store available today, as demonstrated by thorough experiments [56, 192].

The third component of this layer is **SemaGrow** [98], a query processing system that provides a single SPARQL endpoint for federating multiple remote SPARQL endpoints. It is also capable of transparently optimizing queries and dynamically integrating heterogeneous data models by applying the appropriate vocabulary transformations. To boost federated query execution, it employs vocabulary mapping techniques and a balanced query optimizer, considering instance statistics from the federated bases, where available. SemaGrow is highly efficient, consistently outperforming the state-of-the-art in federated query processing [98]. In our case, SemaGrow federates Cassandra and Strabon, offering a unified SPARQL endpoint for both of them to GeoSensor's user interface. In this way, GeoSensor gains in query performance (with respect to other systems, e.g., FedEx and SPLENDID) and has increased extensibility – in case new sources need to be added in the future.

GeoSensor's interface is offered by **Sextant** [468], a web-based application for exploring, interacting and visualizing time-evolving linked geospatial data. Sextant is also capable of

³¹<http://rdf4j.org>

³²<https://postgis.net>



Figure 3.11: User criteria for triggering (a) Change Detection, and (b) Event Detection.

creating, sharing, searching and collaboratively editing maps and of producing statistical charts out of statistically enhanced data sets. It relies heavily on Semantic Web technologies but offers an intuitive interface that allows both domain experts and lay users to exploit all available features. In order to cover all requirements of GeoSensor, Sextant has been extended with three new functionalities:

(i) *Core functionality*. Sextant provides an intuitive interface for initiating the event and the change detection processes of GeoSensor. The window for launching change detection appears in Figure 3.11(a). The user selects an area of interest either by typing its name (with the help of auto-complete), or by highlighting it on the Earth Map. The credentials for Copernicus Open Access Hub are also required along with the reference and the target date. For event detection, Figure 3.11(b) depicts the window that prompts users to define three optional search criteria: an area of interest, a time window defined by two dates, or a keyword that pertains to events of interest. The last criterion can be a combination of location or entity names, or any other words that are likely to appear in an event title. Users can also search for events by setting as the area of interest one that appears in the results of change detection.

(ii) *Authorization/authentication*. To support history over each user’s actions, Sextant implements a sign-up and login functionality. At its core, lies a database located in GeoSensor’s server that holds all account information along with the encrypted passwords. To ensure security over the network, Sextant can be deployed using the HTTPS protocol. When GeoSensor first loads, the user is prompted to create a new account, or to log-in using an existing one. Three types of users that are supported: (a) The *administrators* have full access to all the supported functionality, including the history panel, and are responsible

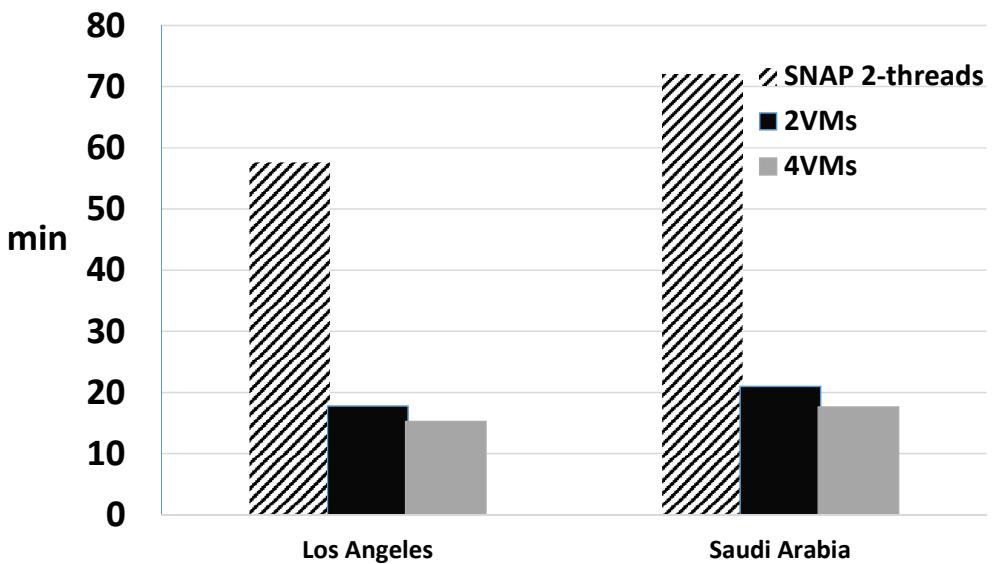


Figure 3.12: Execution times for the different parallelization approaches of the change detection workflow.

for accepting or declining sign-up requests by new users. (b) The *classified users* are the main users of the application and have full access to all the supported functionality, including the history panel. (c) The *unclassified users* are potential trial or occasional users that have limits in using the supported functionality: they lack a history panel, they cannot search for events using keywords, and their event detection searches return up to 5 events. They are also deprived of the "SMART" buttons that alternate change and event detection.

(iii) *Live Twitter keyword search*. To further clarify the map visualization with the latest raw information, overcoming the processing delay of the event detection layer, Sextant offers an embedded Twitter keyword search function that supports all Twitter API filters, such as "#" or "@". Using up to five keywords in the search field, Sextant can fetch / update relevant tweets in descending chronological order, presented via an efficient infinite scroll technique.

3.4.4 Experiments

We now present a preliminary experimental evaluation of GeoSensor's main functionalities, namely the change and the event detection workflows. Note that our evaluation focuses on time efficiency, aiming to assess the response time of each workflow. In other words, effectiveness lies out of the scope of this evaluation, as GeoSensor employs unsupervised state-of-the-art methods for each operation.

3.4.5 Change detection

For change detection, we evaluate the time efficiency of two different approaches: *(i)* the Change Detector, which uses Apache Spark to parallelize the process depicted in Figure 3.8. *(ii)* the baseline approach, which corresponds to the multithreaded implementation of the same workflow, as provided by ESA’s SNAP Toolbox.

Data. As test data, we use two pairs of Sentinel-1A images. One comprising two images of Los Angeles, with file sizes of 508MB and 504MB, and one consisting of two images of Saudi Arabia, with file sizes of 524MB and 526MB.

Experimental Setup. All experiments were performed on a server with Ubuntu 12.04, 132GB RAM and 4 AMD Opteron 6320 processors, each having 4 physical cores and 8 logical cores at 2.80GHz. For the Spark implementation, we created 4 virtual machines (VMs), each one comprising two cores and 20GB RAM. For each pair of images, we used 2 and 4 VMs. In each case, one VM was the master and the rest were used as slaves. The multithreaded implementation of SNAP was run using 2 cores on the same server. For each method and configuration, we took 3 measurements of the execution time and reported the average in Figure 3.12.

Time Efficiency & Scalability. As shown in Figure 3.12, the 2-VMs Spark implementation is three times faster than the multithreaded one. This shows that the communication overhead of Spark is negligible in comparison to the processing time and does not affect the execution times. Furthermore, as we add more slave nodes to the Spark implementation, the execution times decrease consistently. We are working, though, on further improving this performance so as to achieve a linear speedup.

3.4.6 Event Detection

For event detection, we perform an empirical evaluation of the runtime performance of two approaches: *(i)* the Event Detector, which implements the Spark-based distributed similarity mapping pipeline illustrated in Figure 3.9, and *(ii)* the baseline approach, which parallelizes the same pipeline using the Java multithreading library.

Data. We use the Reuters 21K news articles dataset³³. Preprocessing discards everything but the clean text, title and publication date information, storing all data in Cassandra.

Experimental Setup. We run a set of experiments for two different input sizes, namely for input batches of 4,000 and 8,000 articles to be clustered into events. These sizes correspond to approximately 16 and 64 million unique article pairs. For each batch size, we apply the baseline approach using 2 threads, while for the Event Detector we vary the number of Spark partitions $p \in \{2, 4\}$. For each configuration, we perform 5 experiments and compute the mean average execution time. We run all experiments on a single, 8-core 2.6 GHz Ubuntu 14.04 virtual machine with 32 GB of memory. For data storage, we use a Cassandra 2.2.4 docker container.

³³<http://www.daviddlewis.com/resources/testcollections/reuters21578/>

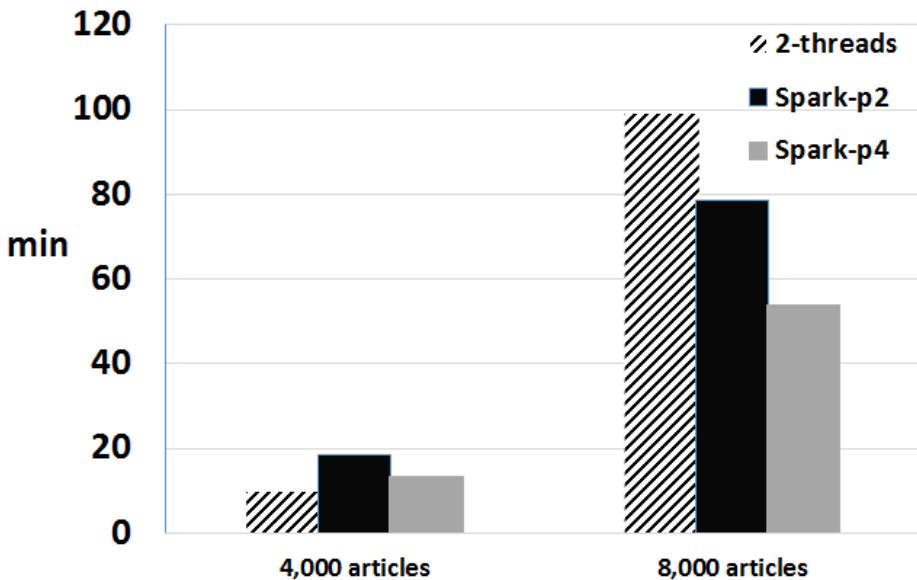


Figure 3.13: Execution times for parallelization approaches of the event detection workflow.

Time Efficiency & Scalability. Figure 3.13 depicts the execution time results per configuration. For the Event Detector, we observe that the runtime drops significantly as we increase the number of Spark partitions, i.e., the number of jobs run in parallel. Yet, the baseline approach is significantly slower only for the largest batch size. The reason is that for a small number of small texts, as in Reuters 21K, Spark’s parallelization overhead is higher than the speedup it achieves. We are working on improving the Event Detector so that its performance is competitive for small workloads.

3.4.7 Conclusions

The work conducted in this study presented GeoSensor, an open-source system we developed as a contribution to the H2020 BigDataEurope project. To the best of our knowledge, GeoSensor constitutes the first system that applies Semantic Web technologies to a combination of remote and social sensing, in an open-source implementation. The RDF data model in particular is crucial for GeoSensor’s functionality, as it offers two major advantages compared to traditional, semantic-free approaches. First, it allows for effectively dealing with Variety, seamlessly combining all data processed by GeoSensor towards meaningful analysis. It also facilitates the use of ontologies together with reasoning techniques so as to derive new facts that are not explicitly expressed in the available data. The second advantage comes from the power of linked open data and semantics. Transforming GeoSensor’s data into RDF allows for effortlessly interlinking it with other data sources and for discovering hidden links between entities that assist in data analysis – a process that not only provides richer data, but also allows building fully automated workflows using machine learning algorithms.

Moreover, GeoSensor can be easily deployed in any cluster. *All its components are provided as Docker images, publicly available through the BDE repository³⁴*, with the whole system able to launch through a single docker-compose file, running the individual components as Docker containers within Docker Swarm³⁵. To further enhance its usability, GeoSensor offers an intuitive UI, suitable for both expert and lay users, despite the rich information it processes. In fact, GeoSensor provides a *hands-off functionality* in the sense that all its operations are fully automatic, requiring no specialized input or domain knowledge from its users. In this way, GeoSensor makes a big step forward in the exploration and visualization of big data in the context of remote sensing. Our preliminary experimental study also demonstrated the high time efficiency of our system.

In the future, we will test GeoSensor in rigorous, operational scenarios where fast, easy-to-use tools are crucial to decision-making.

3.5 Scaling and Enrichment of Automatic Summarization

In this section, we investigate summarization systems and their applications, based on graph-based representations. Additionally, we investigate scalability, entity-based extensions and evaluation schemes.

3.5.1 Introduction and Overview

Automatic summarization has been under research since the late 50's [401] and has tackled a variety of interesting real-world problems. The problems faced range from news summarization [39, 648, 300, 512, 666] to scientific summarization [36, 615, 689] and meeting summarization [166, 466]. More recently, document summarization has moved on to specific genres and domains, such as (micro-)review summarization [464, 196] and financial summarization [271]. The significant increase in the rate of content creation due to the Internet and its social media aspect, moved automatic summarization research to a multi-document requirement, taking into account the redundancy of information across sources [8, 39, 124, 165, 26]. Recently, the fact that the content generated by people around the world is clearly multilingual, has urged research to revisiting summarization under a multilingual prism [169, 200, 539, 623, 648]. However, this volume of summarization research does not appear to have reached a wider audience, possibly based on the evaluated performance of automatic systems, which consistently perform worse than humans [121, 133, 201]. We should note at this point, however, that even summary evaluation itself is a challenging scientific topic [390]. In this chapter, we show how a novel, multilingual multi-document news summarization method, without the need for training, can be used as an everyday tool. We show how we designed and implemented an automatic summarization solution, named NewSum, which summarizes news from a variety of

³⁴<https://github.com/big-data-europe/pilot-sc7-cycle3>

³⁵<https://docs.docker.com/engine/swarm>

sources, using language-agnostic methods. We describe the requirements studied during the design and implementation of NewSum, how these requirements were met and how people evaluated the outcome of the effort. Our main contributions in this chapter are, thus, as follows:

- We briefly study the requirements of a real-world summarization application, named NewSum. We describe task-aware specifications based on user and application context limitations (e.g. device, communication), source limitations and legal limitations.
- We describe a generic, language-agnostic method for extractive summarization, taking into account redundancy constraints. The method needs no training and minimizes the effort of crossing language boundaries, since it functions at the character level.
- We describe an open architecture for responsive summarization on a mobile setting.
- We provide an evaluation of the system based on non-expert evaluations, to represent market applicability of the system.

In the following section we provide some background on automatic summarization to sketch the related summarization research.

3.5.2 Background

Summarization has been defined as a reductive transformation of a given set of texts, usually described as a three-step process: selection of salient portions of text, aggregation of the information for various selected portions, (optionally) abstraction of this information and, finally, presentation of the final summary text [592, 409]. The summarization research community addresses major problems that arise during the summarization process.

- How can one group texts into topics, given a big set of texts of varying topics?
- How can one detect and select salient information to be included in the summary (ideally without training)?
- How can one avoid redundant or repeated information in the output summary, especially when multiple documents are used as input to the summarization process?
- Can one develop methods that will function independently of the language of documents? To what degree can this independence be achieved?

Up to date, many summarization systems have been developed, presented and evaluated, especially within such endeavors as the Document Understanding Conferences (DUC) and Text Analysis Conferences (TAC)³⁶. The summarization community has moved from

³⁶See <http://duc.nist.gov/> and <http://www.nist.gov/tac/> for more information on DUC and TAC.

single-text (single-document) to multi-text (multi-document) input and has also reached such domains as opinion summarization and “trend” summarization, as in the case of NTCIR³⁷. Different evaluations performed in recent years have proved that the multi-summarization task is highly complex and demanding, and that automatic summarizers have a long way to go to perform equally well to humans [133, 132, 131]. A study on how well a system can perform summarization [195] compared two basic methods of summarization: the extractive and the abstractive. In extractive summarization the summarizer forms a summary by selecting sentences from the original texts. In the abstractive approach, which is how humans tend to summarize, the summarizer creates a (mental) abstraction of the information and then composes a new text based on this abstraction. The study showed that extractive summarization has an upper limit of performance. This upper limit in the study was the performance of humans (considered the best summarizers), who applied extractive summarization through simple sentence selection and reordering. Abstractive summarization seems to be able to perform better than the extractive process. In the domain of news summarization, there exist efforts that are publicly available, proof-of-concept systems. Such systems are the NewsInEssence system [512], the Columbia NewsBlaster [169] and the multilingual NewsGist [623]. A number of commercial efforts for products and services related to summarization are currently available. We briefly overview these efforts in the following paragraphs.

3.5.2.1 Related summarization systems and software

Summly³⁸ is a single-document summarizer, applied on web pages and has just recently been embedded into the Yahoo! mobile application. Wavii³⁹ is an application offering a social media view of news integration, by generating a stream of summarized news. It is multi-document, but applied in a single language (English). It was recently bought by Google and is currently unavailable. Another effort is that of iResearch Reporter⁴⁰, which is an English-only summarization solution, provided via a web interface. It is search-based, meaning that it summarizes the results of searches in a structured report. Similarly, JistWeb and JistDesktop⁴¹ are a set of web and desktop based tools that summarize search results into a single document. Ultimate Research Assistant⁴² is a search-based, multi-document, multilingual summarizer. It incorporates no redundancy removal and provides both web-based access and also via a programmatic Application Programming Interface (API). TLDR Reader⁴³ is a mobile application that provides single document summaries on articles and pages. It only works on English texts. TLDR can be also used via an (API). ReadBorg⁴⁴, based on the TextTeaser summarizer, provides

³⁷See <http://research.nii.ac.jp/ntcir/> for more information on NTCIR.

³⁸summly.com

³⁹wavii.com

⁴⁰iresearch-reporter.com

⁴¹jastatechnologies.com

⁴²urast.com

⁴³tldrstuff.com

⁴⁴readborg.com

single document summaries of news in English. The summarization service is provided as a website and also via a web service API. Other efforts and products related to NewSum include EMMNewsExplorer⁴⁵ which is a web-based news aggregator applied to many languages, which however provides no summary – similarly to a variety of aggregators like Google News⁴⁶, Fark⁴⁷ and others. What none of the above solutions provide is a multilingual, multi-document, news clustering and summarizing infrastructure and front-end software, offering an effective glimpse of news, made to suit mobile user needs. In the next paragraphs we overview research works on the summarization subtasks of salience (or importance) detection and redundancy removal, to support the novelty of our n-gram graph based proposed methods.

3.5.2.2 Sentence and information salience

To determine salience of information, researchers have used positional and structural properties of the judged sentences with respect to the source texts. These properties can be the sentence position (e.g., number of sentences from the beginning of the text, or from the beginning of the current paragraph) in a document, or the fact that a sentence is part of the title or of the abstract of a document [156, 513]. Also, the relation of sentences with respect to a user-specific query or to a specified topic [124, 486, 638] are features providing evidence towards the importance of information. Cohesion (proper name anaphora, reiteration, synonymy, and hypernymy) and coherence - based on Rhetorical Structure Theory [411] - relations between sentences were used in [409] to define salience. The idea was that of a graph, where each sentence is a vertex. Vertices in this graph are connected by edges when there is a cohesion or coherence relation between them (e.g. common anaphora). The salience of a sentence, given this graph, is computed as the result of an operation dependent on the graph representation (e.g. spreading activation starting from important nodes). Before studying the graph-based methods any further, we first overview other common approaches to salience detection in summarization.

Oftentimes, following the bag-of-words assumption, a sentence is represented as a word-feature vector, as in [619]. In such cases, the sequence of the represented words is ignored. The vector dimensions represent word frequency or the Term Frequency - Inverse Document Frequency (TF-IDF) value of a given word in the source texts. In other cases, further analysis is performed, aiming to reduce dimensionality and produce vectors in a latent topic space [182]. Vector representations can be exploited for measuring the semantic similarity between information chunks, by using measures such as the cosine distance or Euclidean distance between vectors. When the feature vectors for the chunks have been created, clustering of vectors can be performed for identifying clusters corresponding to specific topics. A cluster can then be represented by a single vector, for example the centroid of the corresponding cluster's vectors [513]. Chunks closest to these representative vectors are considered to be the most salient. We must point out

⁴⁵emm.newsexplorer.eu

⁴⁶news.google.com

⁴⁷fark.com

that for the aforementioned vector-based approaches, one needs to perform preprocessing to avoid pitfalls due to stop-words and inflection of words that create feature spaces of very high dimension. However, the utility of the preprocessing step, which usually involves stemming and stop-word removal, is an issue of dispute [349, 354]. More recent approaches use machine learning techniques and sets of different features to determine whether a source text chunk (sentence) should be considered salient and included in the output summary. In that case the feature vector calculated for every sentence may include information like sentence length, sentence absolute position in the text, sentence position within its corresponding paragraph, number of verbs and so forth - e.g. see [615]. It has been shown that for specific tasks, such as the news summarization task of DUC, simple positional features for the determination of summary sentences can be very promising for summarization systems [132]. However, in other domains or genres these features are not adequate. The example of short stories falls into this type of case, where a completely different approach is needed to perform the summarization [305]: the specific summary type described may be expected to describe the setting without giving away the details or surprises of the plot. In [280], we find an approach where time-aware summaries take into account the frequency of terms over time in different versions of web pages to determine salience. The notion of Bayesian expected risk (or loss) is applied in the summarization domain by [327], where the selection of sentences is viewed as a decision process. In this process the selection of each sentence is considered a risky decision and the system has to select the sentences that minimize the risk. The CLASSY system [124, 123] extracts frequently occurring (“signature”) terms from source texts, as well as terms from the user query. Using these terms, the system estimates an “oracle score” for sentences, which relates the terms contained within the candidate sentences to an estimated “ideal” distribution based on term appearance in the query, the signature terms and the topic document cluster. Different optimization methods (e.g. Integer Linear Programming) can then be used to determine the best set of sentences for a given length of summary, given sentence weights based on their “oracle score”. Focusing on the graph-related literature on multi-document summarization, we visit a number of works that build on graph structures to build summaries. In [638] the authors create a graph, where the nodes represent text chunks and edges indicate relation between the chunks. In that work, the maximum spanning tree of the document graph that contains all the keywords is considered an optimal summary. More recently, the G-FLOW method [111] builds on estimated discourse relations to build Approximate Discourse Graphs (ADGs). The summarizing process then uses the graph to select one from various candidate extractive summaries, maximizing coherence. The candidate summaries are also graded via a regression model of salience (based on ROUGE scores of training corpora) and a redundancy detector (based on information extraction). The result is a summarizer that searches through possible ordered lists of sentences - by applying a stochastic hill-climbing algorithm - to find a summary that contains maximally salient, non-redundant sentences that form a maximally coherent text. In multi-document summarization, different iterative ranking algorithms like PageRank [80] and HITS [314] over graph representations of texts have been used to determine the salient terms over a set of source texts [434]. Salience has also been determined based on the fact that documents can be represented as “small world” topology graphs

[419]. In these graphs important terms appear highly linked to other terms. Finding the salient terms, one can determine the containing sentences' salience and create the final summary. In another approach [241], content units (sentences) are assigned a normalized value (0 to 1) based on a set of graphs representing different aspects of the content unit. These aspects include: query-relevance; cosine similarity of sentences within the same document (termed relatedness); cross-document relatedness, which is considered an aspect of redundancy; redundancy with respect to prior texts; and coreference based on the number of coreferences between different content units. All the above aspects and their corresponding graphs are combined into one model that assigns the final value of salience using an iterative process. The process spreads importance over nodes based on the "probabilistic centrality" method that takes into account the direction of edges to either augment or penalize the salience of nodes, based on their neighbors' salience. In a related study of graph methods for multi-document summarization [524], we see that cross-document structure (via the Cross-document Structure Theory) can be embedded into a sentence-by-sentence similarity graph to enrich available information. Then, node traits such as node grade, clustering coefficient are used to select the most salient sentences across all source texts. In a work by Cai and Li [87], the authors use a mutual reinforcement principle to determine salient sentences in a query-driven summarization task. The idea is that "a sentence should be ranked higher if it is contained in the theme cluster which is more relevant to the given query while a theme cluster should be ranked higher if it contains many sentences which are more relevant to the given query". To apply this intuition on importance propagation, the authors form a two-layered graph. One layer of the graph contains vertices mapped to topic clusters; the other layer contains vertices mapped to sentences. Edges are drawn between vertices weighted by the cosine similarity of the corresponding vector space representations of the vertex items. Two reinforcement-based methods - Reinforcement After Relevance Propagation (RARP) and Reinforcement During Relevance Propagation (RDRP) – are proposed to determine the importance of sentences in the graph. In the MUSE multi-document summarizer system [376] a set of features related to graphs are used as input features to a genetic algorithm. The algorithm, exploiting a training document set, creates a weighting scheme that allows ranking sentences based on the graph (and several other types of) features. The graph in MUSE shows positional proximity between words [377]: nodes are words and edges are drawn between words that are found to be consecutive in the original text. The system is extended to the MUSEEC system [379], which employs a genetic algorithm that learns linear combination of multiple linguistic, statistical and semantic sentence features to rank sentences in a language-independent manner towards extractive salience-based selection. Several, rather recent, approaches heavily rely on vector space models for word or sentence representations, ranging from term-weight vectors to dense, distributed representation schemes. The NTNU architecture [374] ranks sentences by cosine score comparisons of TF-IDF, CBOW or paragraph vector embeddings [347]. Similar approaches mine term inter-similarity by exploiting distributional and co-occurrence information to arrive at similar semantic content [34]. Statistical sentence and word-level features are used in LIA-RAG [507], with a sentence-wise graph clustering approach grouping similar sentences together via the Jensen-Shannon divergence measure. In another approach

[640], the authors employ semantic information infusion via semantic graphs (e.g. Wordnet [441]) and named entity extraction, followed by a post-processing of resulting semantic vectors with PCA. The work of Balikas and Amini (Balikas and Amini, 2015) examine a variety of sentence embeddings learned by deep neural networks, in conjunction with an examination of serial and pooling-based sentence selection strategies, comparing vectors via the cosine similarity. Furthermore, the Sheffield-Trento system [13] implements source linking by computing a quotation score that quantifies the degree of inter-reference, using term vectors along with detected named entity information. Subsequently, instances are compared with a multitude of similarity measures to extract related pairs. Word and character n-gram features are employed in the work of [321], where LDA-based topic modelling is used to produce topic distributions of input source texts. In the work of Xu et al. [675], the authors use an attentive encoder-decoder architecture [110], employing doc2vec embeddings [347] for the encoder, pre-trained on the Gigaword corpus and fed to an LSTM decoder [252]. The authors train each component separately and examine a variety of ways of passing the embeddings to the decoder, reporting sub-par performance of the decoupled model when applied on out-of-domain data. Rossiello, Basile and Semerano [531] aggregate word embeddings to a sentence vector, scoring the latter with respect to their similarity to the document centroid. A subset of the most similar sentences are retained, while simultaneously considering redundancy between adjacently-ranked sentences. Moreover, Vanetik and Litvak [378] apply a compression scheme where sentence itemsets are replaced with an encoding, following a minimum description length (MDL) principle. Sentence ranking is subsequently applied to generate a summary according to a desired length. Li, Mao and Chen [359] tackle content linking by employing word embeddings, using their inner product and Wordnet [441] for sentence similarity extraction. Additionally, they use LDA-based approaches [102] for sentiment mining and argument labelling. The work of Lloret et al. [388] tackle multi-domain summarization from social media and online reviews, using manual polarity annotations, term frequency and noun phrase length for sentence ranking. Summaries are formed based on target length and sentiment constraints, as well as the candidate cosine similarity score to each polarity group. Other works adopt a topic-based analysis, implemented by various approaches available in the topic modelling literature. The AllSummarizer system [23] forms clusters using the cosine similarity on vectors composed of variety of lexical and sentence-level statistical features, while the ExB model [618] extracts topic-central sentences by iterative applications of TextRank [434] on a sentence graph, built by various sentence vector bag approaches (BoW, TF-IDF) and similarity measures (Jaccard, cosine, and a semantic similarity extraction system). Additionally, the CIST system [647] uses LDA-based modelling [66] to construct topics, with the approach additionally utilized towards extracting representation useful for sentiment annotation of the source texts. Another system [640] examines a “topic-focused” approach via PCA projection, where a single salient sentence - with respect to the highest weight on the principal component axis the source sentence feature vectors are projected to - is retained, arguing that such topic-salient sentences will correspond to the transformation eigenvectors, however with limited success. Alongside the above studies on main summarization pipelines, there have been advances in instrumental summarization sub-tasks. Regarding splitting and tokenization, an eval-

ation from Conroy and Davis [120] reports the Porter stemmer-based FASST-E under-performing against the Rosetta and NLTK splitters on multi-document summarization; they additionally highlight the value of hierarchical sentence interleaving, an approach that uses the structure of the document rather than statistical features, for large document summarization. In addition, the utilization of sentiment, i.e. the expressed polarity in the text as a feature in the downstream task of summarization has been widely used. In the work of Tanev and Balahur [34], unigram and bigram counts, sentiment lexicons and domain-specific resources are employed in conjunction with SVM classifiers towards online forum sentiment extraction. Others use LDA-based approaches [647] to arrive at dense representations for sentiment classification, while the approach in [13] employs a feature selection step prior to a support vector regression scorer. The study of Krejzl et al. [321] use maximum entropy classification on a variety of n-gram-based features. Finally, on the summary evaluation subtask, Ellouze, Jaoua and Belguith [159] build a summary evaluation system that employs multiple summary evaluation scores, lexical and syntactic features, concatenated to vectors to represent summaries. The latter are fed into a feature selection post-processing step, the result of which is used to score summaries via a variety of ensemble learning models. In light of the existing body of related studies, within this work we tackle the problems of salience detection in extractive multi-document summarization using a unified, language independent and generic framework based on n-gram graphs. The contributed methods offer a basic, language-neutral, easily adaptable set of tools. The basic idea behind this framework is that neighborhood and relative position of characters, words and sentences in documents offer more information than that of the ‘bag-of-words’ approach. Furthermore, the methods go deeper than the word level of analysis into the sub-word (character n-gram) level, which offers further flexibility and independence from language and acts as a uniform representation for sentences, documents and document sets. As opposed to related works, we do not use centrality traits of the n-gram graph nodes or other graph properties to determine salience. We do not use training or search to rank our sentences. We do not apply propagation of some kind to determine importance. Salience in our system is determined via a set of similarity operators that are applied between topic- and sentence-representative n-gram graphs. The representative graphs are generated via a custom graph merging operator applied on sets of sentence n-gram graphs. The work presented in this book heavily builds upon conclusions and lessons from previous technical reports (e.g. [210]). However, the summarization method described herein has significantly different analysis and steps e.g., for subtopic detection, as well as a different overall approach on segmentation (no sub-sentential chunking), essentially constituting a completely novel method of summarization.

3.5.2.3 Redundancy detection

A problem that is somewhat complementary to salience selection is that of redundancy detection. Redundancy indicates the unwanted repetition of information in a summary. Research on redundancy has given birth to the Marginal Relevance measure [92] and the Maximal Marginal Relevance (MMR) selection criterion. The basic idea behind MMR

is that “good” summary sentences (or documents) are sentences (or documents) that are relevant to a topic without repeating information already in the summary. The MMR measure is a generic linear combination of any two principal functions that can measure relevance and redundancy. Another approach to the redundancy problem is that of the Cross-Sentence Informational Subsumption (CSIS) [513], where one judges whether the information offered by a sentence is contained in another sentence already in the summary. The “informationally subsumed” sentence can then be omitted from the summary. The main difference between the two approaches is the fact that CSIS is a binary decision on information subsumption, whereas the MMR criterion offers a graded indication of utility and non-redundancy. Other approaches, overviewed in [15], use statistical characteristics of the judged sentences with respect to sentences already included in the summary to avoid repetition. Such methods are the NewWord and Cosine Distance methods [343] that use variations of the bag-of-words based vector model to detect similarity between all pairs of candidate and summary sentences. Other, language model-based methods create a language model of the summary sentences, either as a whole or independently, and compare the language model of the candidate sentence to the summary sentences model [713]. The candidate sentence model with the minimum KL-divergence from the summary sentences’ language model is supposed to be the most redundant. The CLASSY system [124, 123] represents documents in a term vector space and enforces non-redundancy through the following process: Given a pre-existing set of sentences A corresponding to a sentence-term matrix MA, and a currently judged set of sentences B corresponding to a matrix MB, B is judged using the term sub-space that is orthogonal to the eigenvalues of the space defined by A; this means that only terms that are not already considered important in A will be taken into account as valuable content. The G-FLOW method [111] uses a triple representation to represent sentences and to determine redundancy across sentences. In this work, we have used a statistical, graph-based model of sentences by exploiting character n-grams. The strategy, similarly to CSIS, compares all the candidate sentences and determines the redundant ones. We use no deep analysis and we function in a language-independent manner, by using the sub-word (character-based) representation of n-gram graphs. The redundant sentences are removed from the list of candidate sentences before generating the summary. In the following sections, we provide the study and details related to our proposed method: we overview the requirements of a real world news summarization system; we discuss the research problems behind some requirements and how a novel method of summarization and an open architecture were devised to provide a solution. We then provide an evaluation of our approach based on user studies and conclude the investigation.

3.5.3 NewSum: News Summarization in the real world

3.5.3.1 Real-world requirements

We saw that, in the summarization domain, a variety of problems arise when attempting to provide human-level summaries. NewSum was our effort to provide a task-specific - or

“full-purpose” as Sparck-Jones put it [294] - implementation of a summarization system with one specific goal: allow humans to get a maximum coverage picture of everyday news in a limited time, avoiding redundancy. The implied process for the generation of summaries in such a system, as was indicated in the introduction, is as follows. First, we gather articles from various news sources, which we then group. The grouping is based on the real events they refer to. We determine the most important aspects of an event. Then we try to detect and extract the most representative sentences covering these aspects to form a summary, while avoiding redundancy of information. We identified several user requirements plausible related to a news summarization application. First, the summary should be provided with minimal delay (ideally in real-time). Second, all the implementation details should be hidden behind a friendly and effective interface. Third, the system should be multilingual to maximize the impact of the application. We decided that another important aspect of our system would be a feature allowing the user to provide feedback on the summaries. This would support error detection and analysis, as well as an estimation of how users (who are usually non-experts in linguistics) perceive the performance of the system. One thing we should stress here is that we decided to follow a strategy which would be penalized in most summarization research tracks (e.g., TAC, DUC): our summaries would not be cohesive, fluent summaries; they would be a set of points. The promise of the system, essentially based on a corresponding specification, is to provide the main points of an event and the user is made aware of this promise. We selected this approach because we assumed that a user expecting a cohesive, fluent, short text summary will be disappointed if he sees an extractive summary (i.e., a collection of extracted sentences). We remind the reader that the assumption is completely in line with the task-specific requirements we have set: extractive summaries may well be suited for a daily news update. Another aspect of the user requirements is related to the provenance of information. When an application consumes and broadcasts content, the application publisher should be aware of the legal problems that may arise: an application cannot claim ownership of the content it uses from external sources. Especially in our case, where the system draws from copyrighted material we wanted to:

- make “fair use” of the provided material.
- point to our source, to allow the user to check the full text and verify the validity of the summary.
- provide a list of all the sources used for an event, even if the sentences used in the summary are only from a subset. This implies that there are cases where different sources have significant overlap of information, and the text from one subsumes the text from the other. This is very common in the world of news.
- allow the user to select a subset of sources for the summaries (based on his preferences).

Given the above discussion, more requirements were added to the original set. Summaries were to be provided as sets of sentences/points. In each such point we should refer to the original source, while also keeping links to all the sources that were used to

describe the event. Finally, we should allow the user to select a subset of the available sources to suit one's needs. We said that the aim of NewSum was not clearly a research aim, but the processing we needed to perform demanded the support of a variety of research domains to build a usable system. In the following section we show how different research domains map to the individual steps of the processing NewSum performs.

3.5.3.2 From n-gram graphs to Markov Clustering

We have claimed that the steps for analyzing content into summaries are four: gather articles from various news sources; group articles into news events; determine the most important aspects of an event; determine the most representative sentences covering these aspects; avoid redundancy of information in the summary. The only step that does not require a research effort is the gathering step (which we will describe in the next section). The other steps are mapped to corresponding research domains as follows: the grouping of articles and the detection of important aspects is mapped to text clustering; the selection of important sentences is mapped to salience detection in the summarization domain; the avoidance of repetition of information is mapped to redundancy removal. We note that NewSum is multilingual, i.e. language-agnostic, which sets an important requirement on the applicable research methods. Previous research [201, 210, 209] has shown that the n-gram graph text representation is a powerful tool that allows representing texts and combining them, comparing them regardless of underlying language. N-gram graphs have, notably, given birth to state-of-the-art summary evaluation methods [205, 203]. In the following paragraphs we review the basics of n-gram graphs and see how they were combined with Markov Clustering (MCL) [634] to achieve text clustering. We also describe how they were an indispensable part of the summarization pipeline, providing - together with the n-gram graph framework algorithms and operators - a generic tool for summarization subtasks.

3.5.3.3 N-gram graphs: the basics

An n-gram graph is a graph representing how n-grams are found to be neighbors, within a distance of each other, in a given text. An n-gram is a, possibly ordered, set of words or characters, containing n elements. The n-gram graph is a graph $G = (V, E, L, W)$, where V is the set of vertices, E is the set of edges, L is a function assigning a label to each vertex and to each edge and W is a function assigning a weight to every edge. The graph has n-grams labeling its vertices $v \in V$. The edges $e \in E$ connecting the n-grams indicate proximity of the corresponding vertex n-grams. Our chosen labeling function L assigns to each edge $e = (v_1, v_2)$ the concatenation of the labels of its corresponding vertices' labels in a predefined order: e.g., $L(e) = L(v_1) + SEP + L(v_2)$, where SEP is a special separator character and the operator $+$ is, in this context, the operator of string concatenation. For directed graphs the order is essentially the order of the edge direction. In undirected graphs the order can be the lexicographic order of the vertices' labels. It is important to note that in n-gram graphs each vertex is unique. To ensure

that no duplicate vertices exist, we also require that the labeling function is a one-to-one function. The weight of the edges can indicate a variety of traits: distance between the two neighboring n-grams in the original text, or the number of co-occurrences within a given window (we note that the meaning of distance and window size changes by whether we use character or word n-grams). In our implementation we apply as weight of an edge, the frequency of co-occurrence of the n-grams of its constituent vertices in the original text. a function assigning a weight $w(e)$ to every edge. We repeat that the edges E are assigned weights of $c_{i,j}$ where $c_{i,j}$ is the number of times a given pair S_i, S_j of n-grams happen to be neighbors in a string within some distance D of each other. The distance d of two n-grams S_i , which starts at position i , and S_j , which starts at position j , is $d = |i-j|$. The selection of a distance value allows different levels of fuzziness in our representation. We note that more in depth analysis of different types of n-gram graphs can be found in the corresponding original paper on n-gram graphs [205].

Here, we will briefly illustrate the process of mapping a string to a character n-gram graph, which is a language-agnostic version of n-gram graphs. Given a string, e.g. “abcdef”, two steps are needed to form an n-gram graph (cf. Figure 3.14):

- First we extract all (overlapping) unique n-grams, e.g. 2-grams and form one node per n-gram. In our example this would be: “ab”, “bc”, “cd”, “de”, “ef”.
- Second, we connect with edges all the n-grams that are found to be neighbors. Two n-grams are considered neighbors, when they are found to be within D characters of each other in the original string. In the example of the figure, “ab” is a neighbor of “bc” for $D=3$, but “ab” is not a neighbor of “ef”.

Once we have drawn the edges, we assign weights to them. The weight of an edge indicates the number of times the two node n-grams were found to be neighbors in the original string (thus, the weight is a positive integer number). In our string all n-grams are found to be neighbors only once. Due to the fact that in this work we look for neighbors in both directions (left and right) the resulting graph has two edges per pair of neighboring n-grams and is essentially equivalent to an undirected graph.

Given this process, we can represent everything from a single sentence to a whole text as an n-gram graph. We note that no preprocessing (stemming, lemmatization, punctuation removal or stop-word removal) is performed on the string. A second way to use the n-gram graphs is to use token (e.g., word) n-grams. In this case some preprocessing is implied, even if only to split the text into tokens. The mapping process is the same, with the difference that distances are measured in tokens instead of characters. Given a set of n-gram graphs, we can apply several operators [199]: the conjunction operator is a binary operator which keeps the common part (edge set) between two graphs A and B . For a common edge (i.e, an edge that appears in A and in B , regardless of its weight) with weights w_A and w_B in the corresponding graphs, the weight in the resulting graph is the average of w_A and w_B . The update operator allows merging a set of graphs into a representative (or class) graph. The merged graph contains all the edges of the source graphs, and common edges in the source graphs result in a single edge with averaged weight in the resulting

graph [202]. On the other hand, several similarity functions have been used to compare n-gram graphs [205]. In this work, we use the Size Similarity (SS), Value Similarity (VS) and the Normalized Value Similarity (NVS) functions. The SS function is simply the ratio of the edge counts of two graphs. Thus, given two graphs G_1 and G_2 , with corresponding edge counts of $|G_1|$ and $|G_2|$, then $SS(G_1, G_2) = \min(|G_1|, |G_2|)/\max(|G_1|, |G_2|)$. We note that to form the ratio we always use the minimum count as the nominator and the maximum count as the denominator. The SS function is trivial, in that it pays no attention to the contents of the graphs, but only to their relative size (edge count). The VS function compares two graphs based on their common edges and also takes into account their edge weights and relative graph sizes. In $VS(G_1, G_2)$, each edge e that is common between G_1 , G_2 and has a weight of w_1 , w_2 in the corresponding graphs, contributes a value of $VR(e) = \min(w_1, w_2)/\max(w_1, w_2)$ to the similarity. If SR is the sum of all the VR values for all the common edges between G_1 , G_2 , then $VS = SR/\max(|G_1|, |G_2|)$. The NVS function is calculated as $NVS = VS/SS$. NVS ignores the relative graph sizes, but takes into account common edges and edge weights. We note that all three similarity functions reported here return values between 0.0 (no similarity) and 1.0 (maximum similarity). In the case where each string is represented with two different n-gram graphs, e.g. a 2-gram graph and a 3-gram graph, one can calculate the Overall Similarity (OS) between two strings S_1 and S_2 , taking into account both levels of analysis. We first calculate the similarities between the graphs with equal n values (2-gram graph of S_1 and 2-gram graph of S_2 ; then, 3-gram graph of S_1 and 3-gram graph of S_2), which gives e.g. V2 and V3. Then, we calculate OS as the weighted average of the similarities: $OS = (2xV2 + 3xV3)/(2+3)$. Once again, the similarity values output from OS are between 0.0 and 1.0, assigning higher importance to higher n-grams (the weighting factor). In the following paragraphs we describe how the above representation and operators – termed the n-gram graph framework – can be applied to face the different research problems NewSum needs to face.

3.5.3.4 Event detection as text clustering

In NewSum we have used two sets of hand-picked sources: one for Greek and one for English news. Each source provided news via an RSS feed, which encodes article information in a semi-structured manner. Each feed was assigned a category label by its publisher. Given this set of sources providing news feeds, we needed to group news items per news category into events. We need to perform this grouping, because a summary makes more sense per event than over irrelevant documents. The topic clustering needs to be a very responsive process, thus we used a mechanism similar to blocking for entity resolution - e.g. see [160]. The idea is that we use a very quick, similarity-based process to perform the clustering into events, however in our case we have a bias to cluster precision than recall. In other words, we do not mind that much if the cluster misses a single text, but we do mind if the cluster contains an irrelevant text. This is related to the fact that an irrelevant text in a news cluster causes problems in all the following steps, while the missing text may simply be found in a separate cluster (possibly alone) and thus no significant loss is conceded. The document clustering has several steps, as follows. First,

we pre-process the text (including the title) to keep only Capitalized words and numbers that appear. This step attempts to perform a very simplistic and high-speed named entity recognizer equivalent, with the addition of a number recognizer. This heuristic implies that the named entities and the numbers are the main identifying information of a news item. The output of the process is a series of tokens which are either capitalized words or numbers. For example, given the title “U.S. tells North Korea new missile launch would be huge mistake”, the resulting series would be (“U.S.”, “North”, “Korea”). We then use word n-gram graphs to represent the series (n takes values in 1,2 and $D=3$). This representation implies that the way entities and numbers are found to co-occur in the above series is important and not simply the series themselves. This approach also helps with noise from the previous step, since the co-occurrence of entities is important and, thus, noise may be isolated (it does not repeatedly co-occur with other entities). Based on the word n-gram graphs, we compare all possible pairs of texts. We use a heuristic rule to connect two texts as referring to the same event: the NVS should be above 0.20 while the SS should be above 0.10. This heuristic (converging to these values through trial and error experiments), is based on the assumptions that we need the texts to be overlapping over a certain degree (NVS threshold), but we also need them to be comparable in size (SS threshold). The success over previously unseen instances given the above values was a cluster precision of over 95% with a cluster recall of about 65%. We note that cluster precision indicates the percentage of texts that were correctly positioned within a cluster (based on the cluster topic). Cluster recall indicates the percentage of the texts that belong to a topic which indeed were assigned to the cluster topic. What this achieved is that we were very strict in our selection of the texts for a given topic: no irrelevant texts should enter. If they did, then the summary would make no sense. The second part of the heuristic related to the size similarity (SS) was inserted to avoid problems of very short texts that appeared to have significant overlaps with almost any other text (due to commonly used words). The result of this step was that texts talking about the same event are connected to each other via a “talk about the same event” relation. The final step is based on the assumption of transitivity of the relation “talk about the same event”. Thus, if A and B “talk about the same event” and B and C “talk about the same event”, then A and C “talk about the same event”. This assumption completes the clustering process, by forming groups of texts, where all texts within a group talk about a single event. Given the clusters of texts, we now need to detect topics and subtopics that form the essence of each event. In the next paragraphs we focus on this detection process.

3.5.3.5 Subtopic detection and representation

In NewSum we consider that an event has several aspects, or subtopics. This approach builds on existing related efforts, exploiting (sub-)topic detection for summarization like [20]. In our approach , we start by segmenting the text into sentences. In order to remain maximally language independent we use a statistical sentence splitter (SentenceDetectorME class of the Apache OpenNLP java package ⁴⁸) to perform the splitting. Our splitter

⁴⁸See <http://opennlp.apache.org/> for more information on the Apache OpenNLP library.

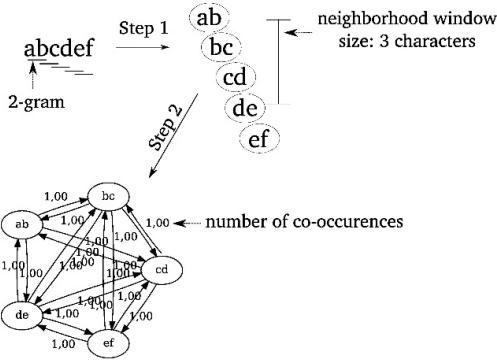


Figure 3.14: From string to n-gram graph

is trained per language and creates a maximum entropy classifier which can determine split-points in a given sentence. We have trained the splitter on both English and Greek texts to support both languages. We continue by comparing all pairs of sentences, based on their character n-gram representation. We use character 3-grams and a neighborhood distance of 3 to represent the sentences. These values have been shown to perform well in a variety of settings [202, 199]. The output of this step is a similarity matrix between sentences, based on the NVS between the n-gram graphs of the sentence pairs. We then apply Markov Clustering - MCL [634] - on the similarity matrix. The result of this process is a set of hard clusters (i.e., no sentence can belong to two clusters at the same time) which we consider as representing different subtopics of an event. Essentially, what we claim is that sentences that are similar to each other in terms of their character n-gram graphs, may talk about the same subtopic. This claim is in agreement with the distributional similarity hypothesis: texts that have a similar distribution of n-grams are likely to speak about the same topic. Since the n-gram graphs represent the co-occurrence statistics of the n-grams in a text and the similarity functions measure the similarity of these statistics, our claim is indeed in agreement. MCL is an unsupervised way to determine automatically – and efficiently – a good number of subtopics, based on the similarities among sentences. This provides an advantage over methods like k-means, which would need an explicit number of topics (k) to work. Furthermore, MCL is very quick to converge, which offers an advantage to several well-known statistical methods for clustering, such as LDA [66]. Given the subtopics, described as sets of sentences that cover the same subtopics, we now need to extract the essence of each subtopic. In the n-gram graph framework, we consider that the essence of a set of graphs is the maximum common sub-graph of all the sets. Thus, to extract the essence of a subtopic, we use the conjunction operator over all pairs of n-gram graphs within a subtopic sentence cluster. In other words, if a subtopic T_i consists of sentences S_1, S_2, \dots, S_n represented by the corresponding character n-gram graphs G_1, G_2, \dots, G_n , and \times is the conjunction operator, then the essence E_i of T_i is: $E_i = G_1 \times G_2 \times \dots \times G_n$. For a whole event, the subtopic detection and essence extraction process results in a set of n-gram graphs, each of which represents the essence of a subtopic. In order to provide the essence of the whole event, we simply need to combine these essences into one representative graph. To this end we use the update operator over all the essences of the subtopics. The resulting merged graph EO is the overall

representative graph of the event.

3.5.3.6 Measuring salience and avoiding redundancy

We presume that a sentence is salient if it is similar to the essence of an event. Going back to the n-gram graph framework, the value similarity (VS) is the type of similarity that takes into account whether a given n-gram graph is maximally similar to another, also using the relative size. In other words, VS is the best choice when we want to take into account the overlap between two graphs (vs. the maximum possible overlap based on the graph sizes, which is reflected by NVS). In order to provide a salience based ordering of sentences, we compare (the graph of) each sentence from the source documents of an event cluster to (the graph of) the essence of the event. Ordering the sentences based on their similarity to the essence, we have a salience-based list of candidates for our final summary. Naively, we could start creating the summary by simply running through the candidate sentences in descending order of similarity to the essence. However, several sentences might talk about the same thing, especially since we are in a multi-document setting. To avoid this problem and tackle redundancy, we perform an a priori filtering of redundant sentences based on the candidate list. The algorithm starts with the most salient sentence S_1 . It compares it to all the following candidate sentences $S_i, i > 1$, by terms of NVS on their character n-gram graphs. If the similarity is above a given threshold (heuristically chosen value in our current setting: 0.3), it means that the later candidate repeats the information of the first candidate and is removed from the candidate list. We iteratively repeat the filtering process for each sentence $S_j, j > 1$, until we reach the end of the candidate list and have removed all redundant sentences. The result of this process is a set of sentences, which maximally cover the essence of a topic, without repeating information. This set of sentences is, for NewSum, the optimal subset of sentences from the original documents that can form an extractive summary. To exploit the results of this process, we created the NewSum application, which provides the infrastructure, the interface and the feedback mechanisms that allow using and evaluating our summarization system in a real-life setting. In the next paragraphs, we present novel modifications of the n-gram graph architecture of NewSum, followed by elaboration of its application details in following paragraphs.

3.5.3.7 Incorporating entity information in n-gram graphs

In the preceding sections, the n-gram graph model is constructed by considering lexical information in the text (i.e. token n-grams and their co-occurrence counts). Other types of information (e.g. higher level concepts, word relations and interdependencies, etc.) are expected to be inferred from n-gram distributional information in the text and learned from scratch from the training data. A straight-forward improvement to this procedure is the direct incorporation of pre-existing human knowledge in the representation, towards aiding the NLP system to handle the task at hand, rather than expecting it to generate everything from scratch. Such human knowledge can be accessed in a structured form via semantic

resources such as semantic graphs, entity databases and various such sources of high-level data. The node-based architecture of the n-gram graph model provides a straightforward extension mechanism towards considering such semantic information units, both in a semantic-only setting and a multimodal scenario where both lexical and semantic information is leveraged in the model. Such an approach is examined in two text clustering settings [622, 504] and includes a two-fold information extraction process. Firstly, TF-IDF filtering retains only the most important terms in the input text, handling tokens with low-information content such as stopwords and other overly prevalent words in the dataset with a variety of strategies (e.g. replacing with a placeholder token or entirely discarding them). Secondly, named entity information is extracted from the text and inserted as semantic metadata objects in the n-gram graph. This results in an n-gram graph with both highly informative terms and named entities, merging lexical and semantic information in a single representation. An experimental evaluation on multilingual article classification (i.e. articles on a fixed topic in multiple languages) from Wikipedia (MultiLing 2015 dataset) as well as news domain data (the 20 Newsgroups dataset) indicates the utility and usefulness of the augmented approach. Specifically, the entity-based modelling improves regular TF-IDF vector space features in terms of F1 score, with the performance improvement being most pronounced in documents rich with named entities that can lead to overlaps. On the other hand, sparse information content does not introduce consistent improvements over clustering results. This finding carries over in the multilingual articles case, where named entity extraction appears to be highly translation invariant, leading to shared entities across multilingual versions of the same text. Finally, the TF-IDF filtering radically reduces the time complexity of the n-gram graph modelling procedure. Additionally, the entity augmentation approach is applied in document clustering formulated as a record linkage problem [504]. There, n-gram graphs provide competitively results to TF-IDF-based representations, and, although the contribution of text versus entity-based information seem to be dependent on the dataset, the n-gram graph approach that leverages both appears to outperform the corresponding text-only or entity-only n-gram graph baselines. This study is also examined in the following section that deals with scalability of the n-gram graph pipeline.

3.5.3.8 Scaling n-gram graph-based analysis

In this section we examine an extension of the n-gram graph comparison pipeline towards achieving better scalability. The extension is applied in two scenarios. First, we look into scalable n-gram graph construction and similarity extraction in supervised text classification. Secondly, we examine improving the pipeline's performance in the context of document clustering, applied in the unsupervised text event detection and record linkage tasks. ARGOT (Apache spaRk based text mininG tOolkiT) is a distributed implementation for n-gram graph operations [320]. It employs Apache SPARK, a cluster computing framework for large-scale data processing [699]. SPARK is a popular and powerful parallelization tool for distributed data processing where operations can be shared in parallel between multiple machines towards improved runtime performance. It has been used for stream-

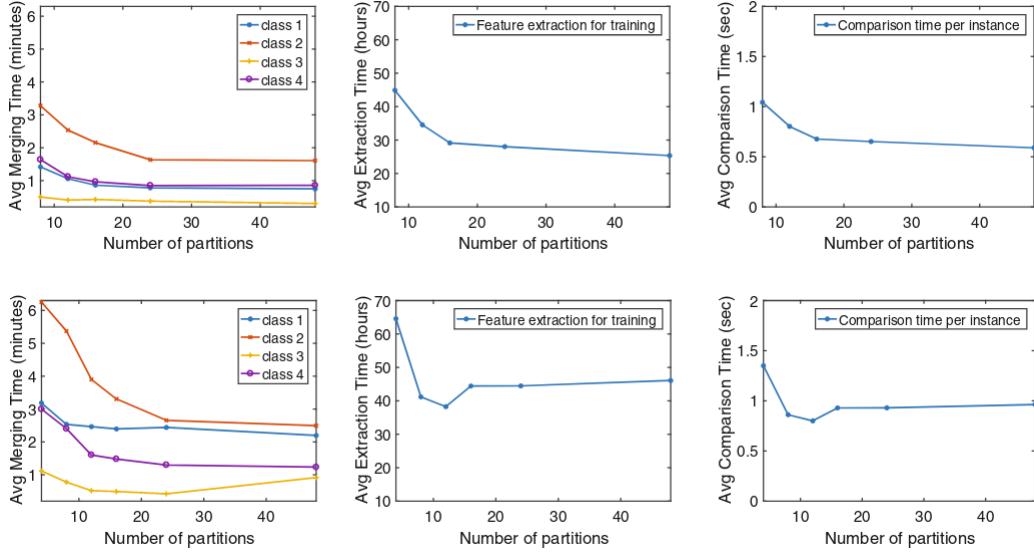


Figure 3.15: Average time elapsed versus the number of SPARK partitions, for experiments run on a single-node machine (top) and a computer cluster (bottom). Columns indicate graph merging (left), feature extraction (middle) and graph comparison (right). Lower is better.

based text processing and text classification [360, 91, 568]. In ARGOT, the authors scale the graph creation process with a mechanism by which graph edges mapping to same vertices are constrained to the same SPARK partition (processing node), thereby improving merging and calculation efficiency of downstream graph operations. The classification process entails building a “class graph”, i.e. an n-gram graph composed of content from training documents for each class label, using instance and class graph similarity scores as feature vectors for learning classifiers. To this end, a graph similarity extraction procedure is implemented by broadcasting the (small) n-gram graph representation of each instance to all partitions, followed by a filtering process to only retain overlapping graph edges, and collect them to the master node for computation of the similarity score. The authors perform a classification and runtime performance analysis on the RCV2 multilingual news dataset, where 10-run average timings are displayed on Figures 4 and 5, representing experiments on a single-node commodity machine and distributed cluster, respectively. On the first setting (Figure 3.15 top), a large scalability capacity is observed, with the algorithm introducing an approximately 50% speedup from 8 to 48 SPARK partitions – especially on classes with many samples (with respect to samples, class 2 > class 4 > class 1 > class 3). The run on the computer cluster (Figure 3.15, bottom) showcases some performance penalties on larger partition numbers with communication overheads counteracting gains from distributed processing.

In order to illustrate the relationship between scalability and the number of instances per class, two subsets of the dataset were created, each comprising 10,000 documents in total, from two and four classes respectively (9,000 training instances and 1,000 testing instances in each subset). The same experiments were performed (10 times per setting

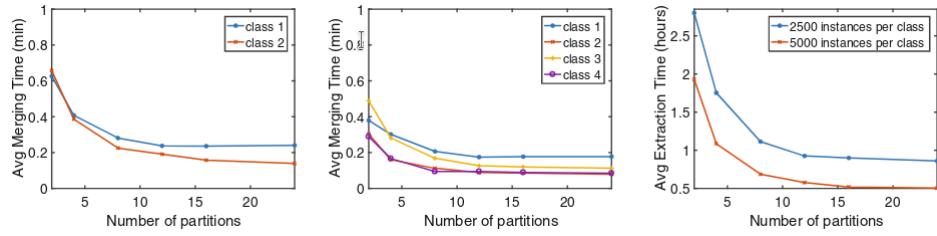


Figure 3.16: Average time elapsed for merging 2 topics (left), merging 4 topics (middle) and feature extraction.

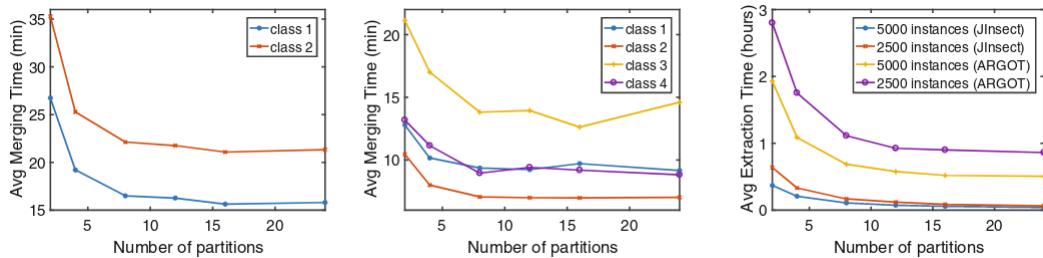


Figure 3.17: ARGOT vs the multithreaded JINSECT implementation

and averaging results), this time using 2 to 24 partitions on the single node setup. Figure 3.16 shows the average merging time per class per experiment, illustrating that merging time depends on the number of the documents in a topic. As illustrated in the average feature extraction time per experiment, having less documents per topic and the same number of total training instances results in longer extraction times. The difference lies in the number of graph comparisons, which are greater in the last case. From this, we can infer that the feature extraction time depends on the number of topics; better scalability of the algorithm can be deduced, as the number of documents (and thus the size of the graphs) increases.

Finally, a comparison of the method to the current multithreaded implementation (JIN-

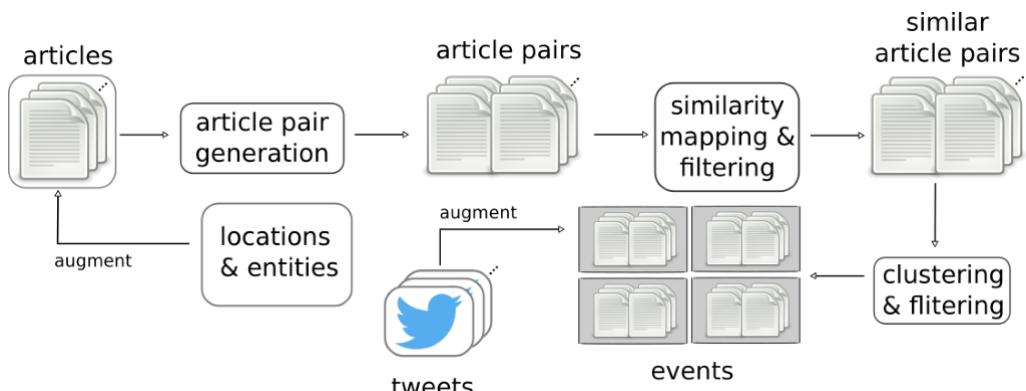


Figure 3.18: The BDE event detection process

SECT toolkit⁴⁹⁾ is performed. Figure 3.17 shows the average graph merging times in the aforementioned subsets correspondingly, as well as the comparison of the ARGOT average feature extraction time to JINSECT. It is evident that ARGOT is faster in graph merging but slower in feature extraction. However, JINSECT fails on the full dataset due to lack of memory scalability, which is not a problem for ARGOT. As expected, ARGOT yields the same results as JINSECT in terms of classification performance.

Regarding n-gram graphs in document clustering, we first examine unsupervised event detection in text [255]. Here, clustering looks for useful structural patterns in the space of the input (i.e. documents mapped to n-gram graph models) and partitions data in meaningful groups (Barlow, 1989). As a result, event detection corresponds to the formation of clusters based on document comparison via appropriate similarity metrics, after some necessary representation transformations. This event detection process is studied as a component of the Security challenge of the “Big Data Europe” (BDE) project [275], and is further refined with the Geosensor system [504, 22]. The component’s task is to form events from news and social media source streams in two steps, from data collection to periodic event formation, enrichment and update. Given the very large volumes of data – harvested for multiple RSS feeds and twitter accounts – the scalability of the event clustering component is critical. An investigation of the time complexity of the procedure identified the document similarity extraction as an important bottleneck of the event detection task. To this end, an implementation of the relevant pipeline was developed in apache SPARK. The event extraction process follows the NewSum summarization algorithm and is based on news documents acquired from the RSS source feeds. First, all unique news item pairs are generated in order to facilitate comparisons of each article with the other. After n-gram graph mapping, pairs of graphs are compared with the NVS operator, with pairs mapped to a similarity exceeding a pre-defined threshold (fine-tuned for the task) are marked as related and retained, while the rest are discarded. The second phase consists of grouping related pairs into clusters, filtered with a support threshold of 2 (i.e. singleton-pair groups are discarded). Subsequently, formed event clusters are enriched with social media items via a similarity-based n-gram graph classification. In addition, news and social media items are augmented with location names, geographical geometry coordinates, named-entities and photographs, extracted from the text, metadata and location names mined from each text. As a result, each formed event consists of a variety of information: news items, social media posts portraying public opinion, location names with geometries and photographs and named entities (e.g. key players that are related to the event). This enhances the utility of formed events towards decision-making as well as automated downstream tasks. Figure 3.18 illustrates the event detection workflow.

One important identified bottleneck of this workflow is the discovery of similar article pairs, where all unique tuples need to be generated from the input article list and compared via the similarity extraction – an operation introducing quadratic complexity with respect to the input list size. We parallelize this procedure with Apache SPARK by transforming the input articles into SPARK resilient data structures (RDDs), a collection of fault-tolerant, immutable data objects distributed across the cluster with each partition processed in par-

⁴⁹Cf. <https://github.com/ggianna/JInsect>

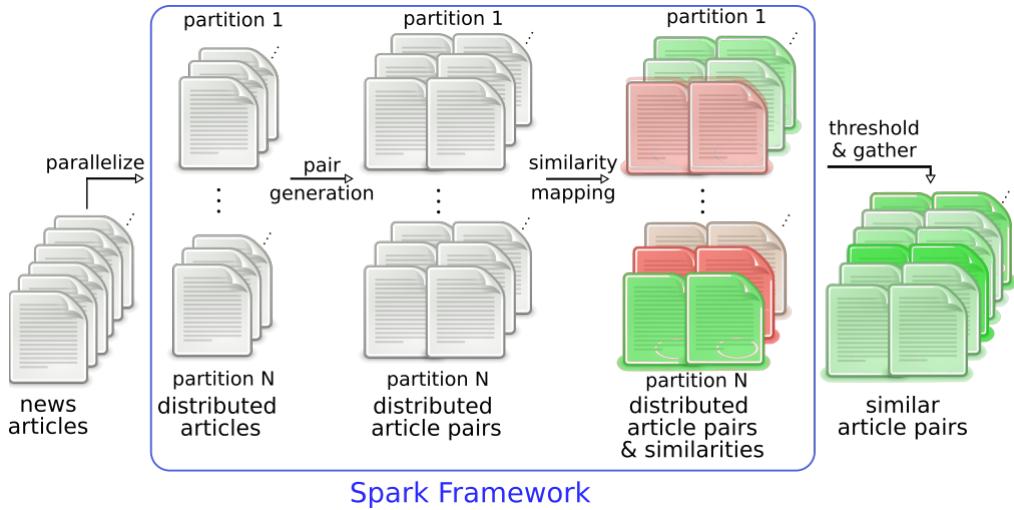


Figure 3.19: The distributed similarity mapping process of the event detection workflow. Green indicates similarity, while red dissimilarity. The color intensity/saturation indicates grades of (dis)similarity.

allel by its host machine. RDD processing is done by transformations like map, filter and reduce operations, which transform, filter and aggregate the data respectively (Zaharia, 2010).

We generate all article pairs by applying a Cartesian product operation on the RDD, followed by thresholding steps that first discard trivial pairs (e.g. singleton identity pairs of the form (x, x) , x being an input article) and second, keeps only one instance out of the two permutations of the same article tuple. This results in $n(n - 1)/2$ combinations for n input articles. Article pairs are subsequently converted into pairs of word n-gram graphs which are reduced to a similarity score using the NVS operator. After similarity threshold filtering, relevant article pairs are retrieved to the host machine and forwarded to the rest of the event detection pipeline. A graphical illustration of the process is presented in Figure 3.19. An evaluation of the distributed event detection implementation was performed, using the Reuters-21578 text categorization dataset, a collection of 21578 documents that appeared on the Reuters newswire in 1987. We used Cassandra 2.2.4 [338] with Docker 1.6.2 [430] for the storage backend and the SPARK 2.11 java API. The experiments ran on Ubuntu 14.04 on an 8-core 2.6 GHz processor with 24 GB of RAM. Regarding the experimental setting, the number of input articles n was varied from 32 to 10000, doubling the input at each step. This resulted in a size of article pairs ranging from 496 to approximately 50 million. SPARK partition sizes were set to $p = 2, 4, 8$ and 16 . A non-distributed multi-threaded run on a single multi-core machine was also performed as the baseline, using all available CPUs. We ran 5 experiments on each (input size, partition size) configuration and the baseline, measuring the elapsed time in seconds for the similarity mapping operation, from the article acquisition from Cassandra up to and including the computation and filtering of the similar pairs. As we are only interested in measuring the time performance of each scenario, we do not evaluate the comparison results themselves.

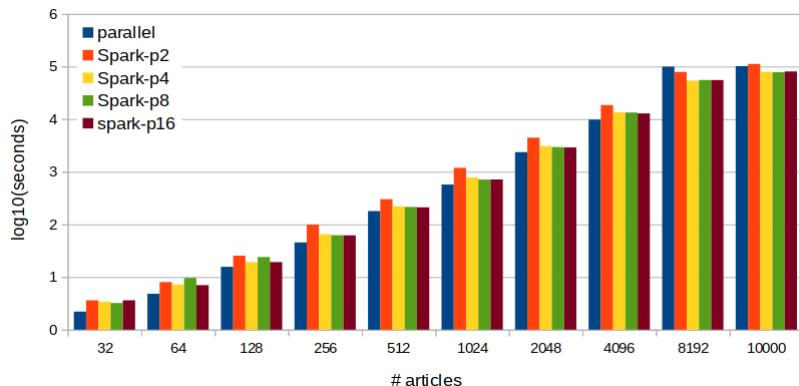


Figure 3.20: Runtime performance results on the Reuters dataset in a logarithmic scale. Lower is better.

Experimental results in terms of elapsed seconds per run are depicted in a logarithmic scale in Figure 3.20. A number of observations can be made. Firstly, the non-linear nature of the problem is apparent, with execution time increasing in an exponential fashion as the number of articles increases. Secondly, the introduction of SPARK is surprisingly detrimental to performance, for an input size less than 4096. This indicates that distributing the similarity extraction process with SPARK for small workloads does not introduce performance benefits; on the contrary, it appears to result in larger extraction times. This deterioration could occur as a result of the SPARK overhead cost being comparable to the distributed execution speedup for smaller article sets, given the computational performance of the n-gram graph algorithm as well as the limited document size in the Reuters-21578 dataset. On the other hand, at an input size of 8192 documents the introduction of distributed execution clearly benefits the running time of the task, with every distributed configuration outperforming the baseline and the partition size of $p = 4$ exhibiting the best performance, i.e. a relative 45.78% increase over the multi-threaded baseline. For 10000 input articles, the SPARK version remains the best performer, with the 8-partition configuration performing best albeit with diminishing gains, resulting in a 23.28% relative performance increase over the parallel run. Moreover, among the distributed execution runs, a partition size of 2 always yields the worst performance compared to other partition settings, for every input size. No clear conclusion can be drawn, on the other hand, for the best performing partition size. In addition, partition sizes of 4, 8 and 16 demonstrate similar performance, for an input size larger than 128 articles. This suggests that the similarity mapping task can benefit from distributed execution for large workloads, with gains however severely diminishing as the number of partitions / workers is increased past a certain point (4, in our case). With respect to configuration rank (last row in the table), the multithreaded runs emerge as the top performing configuration, while the 16 SPARK partition setting is the best performer, on average, despite not achieving the optimal execution time for any experiment.

Given these observations, we observe that performance boosts are attainable in the n-gram graph similarity extraction pipeline using SPARK-based computation distribution.

However, considerable fine-tuning is necessary in order to make sure that the workload, input document size and composition are appropriate to attain such speedups. Future work could examine additional domains (i.e. domains other than news articles), workload sizes or additional parallelization frameworks in order to further establish, extend or revise the above findings. An additional document clustering scaling scenario examined is expressed as a record linkage problem. We use JedAI (Papadakis et al., 2017), a domain-independent entity resolution toolkit that employs a multitude of representation methods and blocking techniques towards efficient record linkage solutions. In the study of Pittaras et al. [504], n-gram graph and a multitude of vector-based representations are employed on text as well as entity-based data, along with a variety of similarity measures for comparing name-value instances towards generation of clusters representing a common category. This setting corresponds to record linkage, in the sense that name-value instances (documents) are aligned to a single real-world entity (clusters). In an experimental evaluation on the 20 Newsgroups and Multiling 2015 datasets, it is observed that JedAI achieves considerable runtime performance improvements (around 63 %), with a trade-off of around 5% of F-measure performance, by introducing data blocking, comparison refinement and pruning schemes of the underlying entity similarity graph. This approach provides another avenue towards accelerating the n-gram graph pipeline, this time by adopting a lossy – but efficient – method of reducing the pool of pairwise comparisons. What all the above experiments indicate is that:

- Summarization based on n-gram graphs can be scaled effectively.
- The scaling strategy is not trivial to determine, since a number of factors (text size, number of texts, genre, etc.) may lead to different data distribution approaches.
- Scaling can also be achieved by applying blocking and other appropriate complexity reduction techniques.
- In a variety of the summarization subtasks, scaling does not appear to strongly affect the effectiveness of the method with respect to the summarization outcome.

3.5.3.9 NewSum: the architecture and the application

We now focus on the NewSum application itself, which integrates the majority of all conducted research into one, free, open-source application, while taking into account real-world requirements. In the NewSum application the main entities are the news article, the news source, the news category, the news feed, the news summary and the news event (or topic). A news article is a time and date annotated piece of news, providing a title and the article body. The annotation is implied in the RSS feed (cf. news feed below) that provides the article. The news source is essentially a news site providing a variety of articles via RSS feeds. The news category is a label describing the broad domain of a news article or a news source. Examples of such categories are: world news, local news, politics, science, etc. In most cases a news source is pre-assigned a category by its editor. NewSum uses this information to create its own news categories. The news

feed is a specific type of web resource (RSS feed), identified by a URL, which provides articles for a specific news category in a semi-structured manner. The news summary is a list of sentences (bullets) related to an event. Each sentence is annotated with its originating news feed. If different source texts contain the same sentence, then either of the sources may appear as the origin of the sentence. The news event (or news topic) is an abstraction of a real-world event. It is described by a set of news articles referring to the event, a title (derived from the most recent news article) and a date. The date of the news event is the date of the latest news article contained in the news event. News aggregators often perform clustering of news articles to form news topic article sets. Having described the main entities in the NewSum application, we overview the architecture of the NewSum system built by the analysis server, the web service and the clients. The NewSum analysis server is the processing backbone of the NewSum architecture. At the server we gather and analyze all the articles from a set of predefined news sources, ending up with the processed summaries. The server periodically queries the news sources for new articles. It performs all the analysis required to order the sentences based on salience but it does not remove redundancy at this point. This last step is kept for the moment when a client requests a summary, because clients can choose their news sources and, thus, redundancy can only be determined after a client requests a summary for a given event for a given subset of the data sources. In order for client software to use the NewSum analysis server output, we provide a web service endpoint (via the Open Source Edition of the Glassfish Server⁵⁰) which simply serves the analyzed, summarized information: the NewSum web service. The endpoint provides the Application Programming Interface (API) for interoperating with the server, getting the detected news topics and the summaries. The web service provides all the required methods to:

- Read the news categories the server covers.
- Get the possible sources that a user can select.
- Read the events (or topics) available at the server, using specific sources.
- Get the summary of a given event, using specific sources.

The NewSum web service makes sure that all the details of the analysis are hidden from client applications and that information is provided at a per-request basis. This latter strategy is meant to minimize network load and latency and allows more flexibility to the client and lower overhead to the server. Furthermore, the fact that the web service is independent of the analysis server allows better workload management and minimizes the impact of one subsystem to the other. Finally, it fulfills the requirements related to speed and responsiveness, since all the data that are sent are *a priori* available (provided as the output of the execution cycles of the analysis server). To provide a friendly user interface, independent of the underlying infrastructure, we have created different NewSum clients corresponding to different settings. The main client is the Android client, which can be used on portable and mobile devices. In collaboration with the NewSum community, we are also implementing web-based versions of NewSum, as well as a variety of

⁵⁰See <http://glassfish.java.net/> for more information on the Glassfish server.



Figure 3.21: NewSum snapshots: (a) The main screen with the topics list for the current category (b) The summary screen, where sentence snippets and sources of the summary are accessible

widgets aimed at desktop and virtual desktop settings (e.g., Windows widgets, KDE widget, iGoogle widget). The clients are built upon client libraries that facilitate developers in their application building effort and boost the reusability of the system. In Figure 3.21 we provide two snapshots of the current version of the NewSum Android application. In the first part we show the basic category interface, illustrating topics pertaining to the selected category, as well as buttons for switching categories and accessing settings and help options. In the second part we show how NewSum renders summaries as lists of sentences, also providing the referring link back to the source document on the Web. The reader should also note the rating button that provides simple-to-use evaluation (1 to 5 stars) for the summary viewed, as well as an option to share the summary through social media.

An early version of NewSum⁵¹ is available as an open source project, building upon the well-established JInsect framework of n-gram graphs (Giannakopoulos, 2010), which is also an open source, free project. It is an effort completed by SciFY⁵², with the research support of the SKEL Lab of NCSR “Demokritos”. The Android application, as well as some web-based preview variations, are available via the SciFY website. Concerning the performance and computational requirements of the summarization method over real data, we note the following: for a total of 110 sources (news feeds) over 18 categories the running time of the whole summarization pipeline (gather, cluster, extract summary) at a server with 8 CPU cores (Intel Xeon at 2.2 GHz) and 12GB of system memory is less than 2 minutes. The implementation is fully parallelizable, scaling with more CPUs. Both the application and the research behind NewSum make more sense if enough people find its results usable. To determine whether such summarization software makes sense, we conducted evaluations on two main aspects: one on the application usability and one on the summary quality, using the feedback mechanics of the software itself. We elaborate on these evaluations in the next paragraphs.

3.5.3.10 Evaluation of Summaries

NewSum had two main questions that needed an answer:

⁵¹See <http://www.scify.gr/site/en/our-projects> for more information on the NewSum project.

⁵²See <http://www.scify.org> for more information on SciFY.

- What do people think about the summaries provided?
- Does the use of NewSum (or a similar application) facilitate reading news, by providing global information from various sources?

We conducted three independent studies to answer these questions. Two studies (one preliminary and one more advanced) were meant to answer the summary quality question. The third was meant to answer the question of whether NewSum serves its purpose. The first user study related to summary quality was conducted during an “open beta” phase of the application. The “open beta” took place between January 2013 and March 2013. During this phase 18 volunteer beta testers were asked to use the program and provide feedback on the summaries they read. The grades were assigned using a 5-star scale: the 1-star rating was mapped to a value of “unacceptable”, while the 5-star rating was mapped to a value of “excellent”. The feedback we gathered contained 119 different summary ratings. The distribution of grades over all the summaries, from all the users is illustrated in Figure 3.22. No information was kept related to who sent the rating and thus user bias cannot be determined in this preliminary dataset. The per language performance - 88 instances for Greek and 31 for English - is similar to the overall distribution of grades, with the Greek average rating having a value of 3.89 (and standard deviation of 1) and the English average a value of 3.55 (with a standard deviation of 1). A Kolmogorov-Smirnoff test (preferred over t-test due to the abnormality of the distributions) showed that we cannot reject that the two distributions (Greek and English grades) are derived from the same distribution. In other words, the two distributions appear very similar. In the rating, grade 1 was meant to indicate “useless or nonsensical summary”, grade 3 was mapped to “Acceptable” and grade 5 to “Excellent”. What is very interesting is that the percentage of summaries with a grade of 3 or higher is almost 87% of the total summaries. This showed that, even though there is space for improvement, most of the times the summary is at least usable (or much better). Moreover, 2 out of 3 summaries were graded with 4 or 5 (“very good” or “excellent” grades).

In the second study, which started within the CICLing 2013 conference, a newer version of the NewSum application, as well as a website were used to conduct a second experiment. This time an anonymized, persistent user ID was assigned to each participating user. We meant to measure the individual bias of users towards higher or lower grades. Figure 3.23 illustrates the distribution of grades over all languages and users.

An ANOVA test indicated that the user is indeed a statistically significant factor related to the summary grades assigned (F -value: 4.162, p -value: $< 10^{-6}$). The language was similarly highly statistically significant, and this was also shown by the average performances: for Greek the average was 4.14 (with a standard deviation of 1.07), while for English the average was 3.73 (with a standard deviation of 1.34). This showed that fine-tuning may make sense on individual languages. In both languages the average performance was good, with more than 90% of the summaries having an acceptable (or better) grade for Greek and more than 80% for English. We stress that Greek and English news do not appear simultaneously in the interface (a program option changes the language used in the application). Of course, we could not test where the users originated from and thus

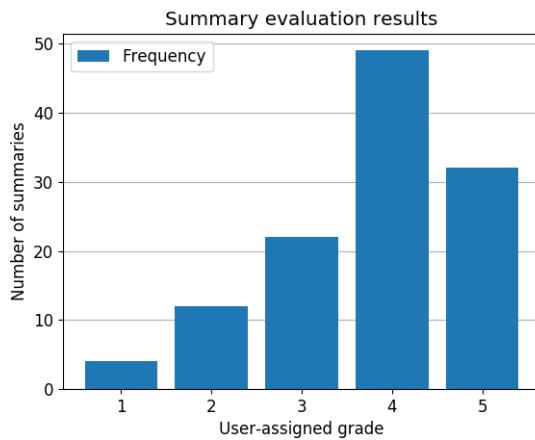


Figure 3.22: Preliminary “Open Beta” summary grades (119 ratings: 31 for English, 88 for Greek)

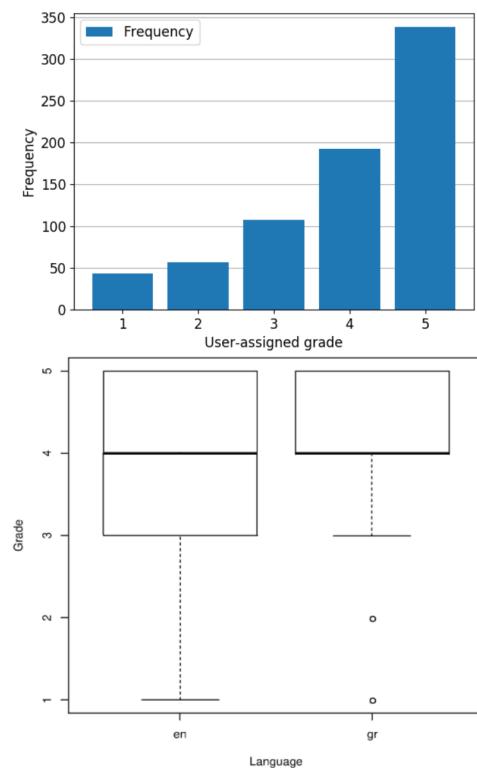


Figure 3.23: User-aware evaluation summary grades (720 ratings: 267 for English, 453 for Greek)

part of the bias may relate to the fluency of the users related to one of the languages.

As a result of the deliberately open experiment setting, users varied heavily as related to their contribution: 58 users contributed ratings, with an average contribution of about 12 ratings each, but the median was between 6 and 7 comments, while a single user provided almost 120 ratings. We tried to remove the user with the 120 ratings, in case he dominated the ratings, but there was slight change in the mean evaluation numbers (-0.01 in the Greek case and -0.15 in the English case). We also checked via ANOVA whether the source of the data (mobile application or website) was important for the grading and we found out that it was not statistically significant for the evaluation. Thus, the interface did not cause a bias related to the perceived quality of a summary. Overall, our evaluation showed that the users think highly of the summaries NewSum creates in both Greek and English. We should however note that applying the same evaluation at large scale with more information per user, would significantly help detect and remove biases from specific user backgrounds and groups. We are also planning to allow richer feedback in the next versions, to be able to perform better error analysis and also allow our system to learn from user feedback via machine learning. To answer to the question of whether a summarization application facilitates news reading, we performed a small scale user experience experiment, limited to 18 Greek and 7 English beta testers. These testers, who were recruited via an open call, were provided a questionnaire that measured different aspects of user experience. The question we will discuss here was expressed as “The use of NewSum allowed me to get informed of the latest news more globally and thoroughly than before”. The answer allowed a 5-scale response from “I totally disagree” to “I totally agree”. 11 out of 18 (61%) Greek users and 4 out of 7 (58%) English users have an answer of 4 or 5 to this question. Only 1 user per language thought that NewSum did not really help (2, in the 5-scaled response). The mean grade for Greek was 4 with a standard error of 0.24; for English the mean was 3.86 with a standard error of 0.46. Thus, these preliminary results indicate that users tend to believe that NewSum can improve their news reading, fulfilling its purpose.

3.5.4 Future Research Directions

What is obvious from our research so far is that summarization can be a useful tool. It appears that optimizing existing algorithms to different needs is the way to engineer generic research tools into usable applications, with a societal and commercial impact. Projects like the NOMAD project⁵³, harvesting information from a variety of sources and aiming to provide actionable updates to decision makers, are examples of the value summarization can bring into this exploding user content generation era. Summarization is essential due to the increase of content. The applicability of summarization will lie in its ability to gather possible millions of sources (texts or other types of data) and combine them into coherent summaries. Such an approach implies research efforts towards a variety of domains as follows.

⁵³See <http://www.nomad-project.eu> for more information on the NOMAD project.

- Language-independent summarization: How can we provide summarization infrastructures over languages that have not been tackled so far, but have millions of speakers and writers (e.g., Chinese, Hindi)?
- Sentiment and argument summarization: Summarizing news has been the focus of numerous research efforts. However, there are many more types of information that would benefit from summarization. Summarizing sentiment and arguments can be of critical importance in domains like policy modeling and business intelligence, since it can provide actionable feedback for decision support. But how can we achieve the summarization in such complex domains, and indeed provide summaries that make sense and can be used as social evidence?
- Summarization as a scientific tool: How can we summarize research effectively, to support such domains as bio-informatics, where the production of papers and studies is difficult for researchers to cope with?
- Holistic summarization: How can we combine different types of content (unstructured, semi-structured, fully-structured) into fused summaries of information? Such summaries could provide invaluable feedback for decision support, without the need to skim through completely different types of data to get updated on facts and numbers. Ongoing efforts in the MultiLing community try to focus on such difficult settings, for example the financial narrative summarization task⁵⁴.

Several ongoing efforts have been fueling the research in these domains and much more is to be seen in the foreseeable future. What we would like to add to this set of research efforts is:

- an effort that would focus on the generalization ability of existing summarization systems and results to new domains and languages.
- an effort that would allow us to measure the extrinsic value of summarization systems from the stakeholders: the users.

NewSum is, for us, a milestone towards these two dimensions of summarization system research.

3.5.5 Conclusion

In this chapter we presented a multilingual, multi-document summarization method, implemented within a real-world setting. We discussed user and system requirements related to a news summarization application. We showed how existing research, mostly based on n-gram graphs, can support and implement a summarization system that covers different languages – namely Greek and English. We overviewed an effective system, called

⁵⁴See e.g. <http://multiling.iit.demokritos.gr/pages/view/1648/task-financial-narrative-summarization>

NewSum, which provides a full infrastructure and user endpoints for the exploitation of summarization on texts provided by real news providers. We described how user studies indicate that summarization is a feasible and useful way to facilitate users in everyday news search. We also described two studies indicating the performance of the presented system, from the viewpoint of (non-linguist) users, i.e. the market setting. We then highlighted how NewSum moved to a larger scale, through the study of alternative scaling strategies, with promising results. We learnt many lessons throughout our effort. First, summarization is useful and can help people, starting today. Second, there exist ways to be language-agnostic in one's methods of summarization. The n-gram graph framework is an effective method to begin with, offering an efficient way to analyze, represent and act upon texts for summarization purposes. Third, we learnt that open, per task evaluations can help judge the usefulness of an approach in a real setting. Integrating response mechanisms in applications and using user feedback may be a good way to perform such an evaluation. Fourth, our open source effort showed that openness allows combining and improving systems, reusing expertise and promoting common experience on a subject. NewSum was built on open source principles, using open source software and will be provided as an open source application itself. Finally, open architectures allow combining different components of summarization under a unified interface. This leads to an implied proposal: why not use summarization applications as a means to evaluate underlying summarization methods? As future work, we plan to apply the proposed methods on corpora of multi-document, cross-/multi-lingual summarization (e.g. from the MultiLing workshops) to test the cross-language applicability. We also plan to incorporate weights for different sources to support different levels of confidence for different sources. Summarization is an indispensable tool for the information overload era. Researchers should also heed popular needs related to summarization and provide the tools and infrastructures to quantify these needs and accordingly evaluate existing systems. This is the main way to large-scale adoption of summarization and to the full exploration of its potential.

3.6 Classifying Videos with Multimodal Deep Neural Networks

In this section, we shift our attention from textual data to the visual and audio modality, in the context of the video classification task, tackling categorization in a multimodal setting with deep neural networks.

3.6.1 Introduction

Multimedia content has become abundant and ubiquitous in our daily lives, resulting in increased consumption and availability of video data [105]. A significant contributor to this has been the surge of popularity of the Internet and advances in video compression techniques [151], enabling digital and interactive television, compilation of large digital video libraries as well as video on-demand services and platforms. The latter, combined with the proliferation of camera-equipped smartphones, IoT and embedded camera systems

have disentangled video consumption from homes or personal computers and encouraged acceleration of video content production on a massive scale.

3.6.1.1 Motivation

The large amount of video content prevents efficient handling and processing by humans alone, necessitating development of automated tools to aid organization and browsing of large volumes of video data. Such operations can be assisted via efficient video annotation and classification systems that assign predefined meaningful labels, categories and semantic indexes that correspond to video content or metadata [135, 301]. Videos are multimodal objects, i.e. each data instance contains multiple exploitable information channels [322, 231], such as the visual and audio components, and are organized in a sequential nature imbued with temporal interdependencies and relationships. Generally, different modalities do not contain identical information content; a video classification system should thus utilize content from all sources and apply information extraction to multiple modalities. Additionally, multimodal information should be combined / fused efficiently, highlighting complementarity, preserving information and eliminating redundancy between modalities. This intuition is the basis of this study, which seeks to investigate and exploit the multimodal nature of video data towards aiding classification.

3.6.1.2 Problem definition

We approach the task as a machine learning problem, specifically single-label classification: given predefined labels $C = \{c_1, c_2, \dots, c_c\}$ and an input video-label tuple (v_i, c_i) , we seek a function $F(v_i) = \hat{c}_i \in C$, with the prediction converging to the true label ($\hat{c}_i \rightarrow c_i$) in the course of data-driven training. We model $F(\cdot)$ with deep neural networks [556] and softmax classification [221]. The contribution of temporal information is investigated via a) fully-connected (dense) feed-forward networks, paired with aggregation mechanisms to fuse sequence instances to video-level objects, and b) sequence-aware approaches that directly model the temporal dimension (i.e. video *duration*) in a refined manner. The intuition behind these two approaches is to compare classifiers that give radically different focus to the temporal component with respect to classification performance. The contribution of the visual and audio modality (which are common channels utilized in the video classification literature [585]) are also examined, by investigating single-modality and multimodal fusion workflows. The two axes of this study are realized in the proposed approaches examined in section 3.6.3

3.6.1.3 Structure

The rest of this section will be structured as follows. In section 3.6.2.1 we present related single-modality approaches and recent work for video classification, with an examination on multimodal approaches laid out in section 3.6.2.2. We include methods that enforce a

multimodal consideration of the video input, using visual, audio, temporal or multimodal approaches to the learning process. A description of the proposed method to address our stated goals follows in section 3.6.3. There, we begin with an introduction to classification, neural networks and deep learning before moving on to the presentation of our proposed workflows, namely the feed-forward and sequential deep neural models in section 3.6.3.2, and the multimodal fusion approaches in section 3.6.3.3.

What follows is the experimental evaluation. We present this in section 3.6.4, describing the datasets and experimental setup in section 3.6.4.1. The main experimental setting, results and discussion with respect to the state goals of the study can be found in sections 3.6.4.2 and 3.6.4.3. We will conclude by a summary of the contributions of this study in section 3.6.6.2, along with potential future work that could complement and extend the investigation of this project.

3.6.2 Related Work

In this section we present related work in the literature for video classification. The reader is encouraged to refer to surveys (e.g. [79, 463, 670]) for a more detailed comparative recounting.

3.6.2.1 Single Modality Video Classification

There has been significant research effort for video classification, exploiting visual, temporal, audio, as well as textual information and metadata, to arrive at a class prediction. Here we outline related work pertaining to single-modality methods that focus on a single information channel of the video.

Visual-based approaches analyze the contents of video frame, looking for useful structure and information expected to aid discrimination. To this end, advances in image classification and content-based image retrieval are exploited for frame representation, followed by aggregation methods.

In [283], the authors modify DCNNs, extending the convolution kernel to the temporal dimension. Their model extracts spatiotemporal features by applying convolution, subsampling and pooling operations to separate channels, i.e. temporally neighbouring contiguous video frames. A final representation is constructed by feature combination of responses from all channels. Experimental results on video action recognition on TRECVID [584] and KTH [560] datasets show that the proposed approach outperforms shallow feature related approaches. In [303], the authors experiment with different ways to introduce temporal information to DCNNs, namely a single-frame model, a late fusion of frames with a fixed temporal distance and two variants of early frame fusion (namely early and slow fusion, with the latter propagating marginal frame information in a slower manner). Experiments on UCF-101 and Sports-1M datasets show that the approach is “not particularly sensitive to architectural details of the connectivity in time”. The slow early fusion variant

consistently outperforms its competition. They note that the single-frame baseline showcases a very strong performance, which hints to either local motion information not being exceptionally important for classification of these datasets, or that a more detailed handling is required. Donahue et al. [154] apply DCNNs for feature vector generation, feeding collections of frame vectors to an LSTM for video classification. Compared to a single-frame softmax classification with voting aggregation, the experimental results on the UCF-101 dataset show the sequential LSTM approach faring better in terms of accuracy. In [695], Ng et al. examine DCNNs with a variety of spatial and temporal pooling approaches, as well as an DCNN - LSTM combination as above. Their work focuses on processing long video clips (reporting processing up to 120-frame sequences). Experiments on UCF-101 show considerable (approximately 10%) improvement to previous approaches that do not utilize motion information. However, they stress the latter is necessary for benchmarks like the UCF-101 dataset for achieving state of the art results.

Audio-based approaches isolate the audio content of the video, followed by the extraction of useful audio features for classification.

Many approaches use low-level audio features; in [387, 386], the authors use a variety of low-level statistical global audio features to distinguish 5 generic television program categories, using HMM classification and ISODATA clustering. In [527, 526], the authors use MFCC features [717] with a GMM for video genre classification. Audio statistics are employed in various works [179, 229], usually alongside additional visual and metadata features. The authors in [674] employ various handcrafted features in conjunction with a HMM for audio-based video event detection on horror and comedy videos, achieving high precision and recall scores on a specialized dataset. Lee et al. [351] use a variety of representation methods (PLSA, Gaussian standard and mixture models) on top of MFCC features, classified with SVMs with various distance functions. They report good performance on a manually obtained and annotated dataset of 25 diverse semantic concepts, with respect to average precision.

A common technique is to convert the audio content into a visual representation and move on with image-based models and analysis. For instance, conversion to spectrogram images, a two-dimensional time and frequency representation, has been popular. In [245], Hinton et al. use layer-wise unsupervised pre-training on stacked RBMs [246] via constructing a generative model of the data, using resulting weights as a good starting point for learning the task at hand via a DNN. Multiple evaluations on a number of datasets on speech recognition illustrate the robustness of the deep models employed. The authors in [127] tackle speech recognition by fitting DNNs, pre-trained via unsupervised training of a deep belief network [459, 249] on the input data, using HMMs to model temporal dependencies and MFCC to represent raw audio. Experimental evaluation on the Business Search dataset [5] show the deep approach outperforming baselines with GMM-HMM modelling, achieving up to 9.2% absolute improvement in terms of accuracy. Convolutional DNNs have been used for audio analysis, often processing visual representations of the audio content in the form of spectrogram images. The authors in [350] use convolutional deep belief networks to learn deep features which they demonstrate have some correspondence to phonemes. They apply their representation to genre and artist

classification, reporting similar classification accuracy to MFCC features. A similar approach is undertaken in [705] for the genre classification task, using convolution, pooling and projection layers to generate deep audio features in a manner similar to the function of the visual cortex [267]. DCNNs have been used in multiple other tasks, such as in [554, 553] for musical onset detection, where they achieve comparable performance to bidirectional RNN models in terms of F-measure, or in [266, 484] where they showcase very good performance in speech emotion recognition. The overview in [144] presents speech recognition advances, emphasizing the improvements introduced by DNN-based filter bank architectures over HMM / GMM-based approaches, as well as the importance of automatic feature learning on audio spectrograms over engineered features like MFCC. Deep architectures are shown to outperform contemporary baselines with respect to word error rate, with additional gains reported with adaptation, transfer learning and regularization techniques. In [225] the authors augment spectrogram DCNNs with “lag matrices”, a self-similarity feature capturing temporal correlations in the data sequence using DCT. Their approach is reported to capture audio structure effectively, producing state of the art results for the music boundary detection task in terms of f-measure. In [107], the authors analyze DCNNs weights learned for music genre classification on short-time Fourier Transform spectrogram inputs, extending the deconvolution approach in [703] for audio data. In [242], the authors tackle multiple instance learning by generating deep features from a three-layered NN with 500 hidden units per layer. The features are fed to multiple neural classification models and experiments on the Audioset dataset [194] show that DNNs softmax-based attention works best in terms of mAP, AUC and d-prime scores, compared to DNN and RNN pooling approaches. In addition, DCNNs have been successfully used for audio segmentation and structure analysis [632], as well as used for audio representation extraction [149], achieving comparable results to using spectrogram image features. The authors in [541] employ LSTM models examining a variety of architectures, as well as introducing a recurrent projection layer modification to alleviate the large number of learnable parameters [542]. In other approaches, the authors [551] have “examined extreme learning machines”, i.e. neural networks with capabilities of analytical rather than incremental learning on a variety of audio-related classification tasks, reporting similar or higher performance to conventional feedforward NNs but significant training time decrease. Furthermore, a dynamic programming approach is proposed in [499, 500] for the binary classification task of music-speech discrimination. In addition, the authors propose a deep learning approach with Restricted Boltzmann Machines in a semi-supervised learning scheme in [501], using MFCC and Fourier Transform-based low-level features and reporting satisfactory generalization performance.

Apart from audiovisual features, temporal qualities exploit the changes occurring within a video clip. Such shifts can occur either by motion of objects of interest in a scene (e.g. a car drives by, a face moves) or by more drastic changes occurring by camera movement or shot change (e.g. a news graphic appears in a news segment, or a cut occurs in a film). In [313] the authors devise a descriptor to capture spatiotemporal information using on 3-dimensional gradients in the video visual stream. In [573], video “tomographs”, e.g. one-dimensional cross-cuts in the temporal dimension, are extracted, to be used as visual input that spans the temporal duration of the video. Optical flow or image velocity

[256, 38] is an estimate of temporal changes in a sequence of frames, approximating the two-dimensional motion field from pattern trajectories in the frames by examining relative positions of pixel intensities. Optical flow has been widely used and achieved state of the art results, with respect to temporal features. In [613], the authors utilize a Hidden Markov Model (HMM) [265] on short video segments in order to model the underlying temporal structure. They evaluate their approach on the event detection task on Trecvid MED data, reporting significant average precision gains over the related work. In [650], the authors use SURF descriptors and dense optical flow trajectories [649] for video action recognition in multiple datasets, while in [277] adopts a decomposition strategy, partitioning motion to dominant and residual parts. In [653] optical flow is used in conjunction with DCNNs on UCF-101 and HMDB51 datasets, fusing DNN and trajectory responses via a spatiotemporal aggregation and pooling schemes. The authors in [283] use DCNNs modified to perform temporal convolution on the human action recognition task, outperforming shallow feature approaches. Yamato et al. [679] use temporal information in human action recognition in frame sequences, by computing mesh features on thresholded visual data and training a HMM for each class. In [272], the authors compute motion features and project them to a one-dimensional signal which is used to train HMMs. They report in binary TV program genre classification. The authors in [179, 621] exploit motion features of the camera (e.g. panning, zooming and cuts), as well as segmented object motion for TV program categorization. A similar approach is investigated in [526], where flow features are computed by tracking pixel-wise frame difference. Another approach is to use recurrent networks to capture information regarding the temporal component inherent in video content directly [595, 695, 154]. For a comprehensive review, see [670]. In [595] the authors use an LSTM network as an unsupervised sequence encoder, compressing video frames or DCNN-produced frame representations to a single vector representation. The authors report modest improvements, when applying the model to supervised classification on the UCF-51 and HMDB51 datasets.

Other approaches use text metadata that often accompany a video. For example, dialogue transcripts, subtitles, or hearing-impaired captions that contain a documentation of sound effects in the video. In addition, semantic information like tags and partial categories may be available. This textual information can be subsequently used with a text representation model (e.g. bag-of-words vectors [544]) as discriminatory features for classification. Textual information can be extracted if not available in metadata – for example, Dimitrova et al. use an OCR-based text box detection and understanding [150], where text elements in video frames are mined via an image processing approach. Furthermore, the authors in [261] apply a text-based approach with news videos on YouTube, using user video titles, descriptions and comments for video categorization and recommendation.

Finally, some approaches incorporate existing domain-specific knowledge to aid discrimination. In [447], expert knowledge in sound energy dynamics in film is exploited in horror film binary classification. In addition, face detection and tracking is a popular high level feature. In [150], the authors apply a face detection and tracking procedure along with OCR-extracted text, using a HMM model for TV program categorization. Face recognition features are used for news videos classification along with text and multimodal features

in [654], using SVMs and GMM learning models.

3.6.2.2 Audio-visual fusion

Regarding multimodal fusion methods, the authors in [711] focus on the audio component, conducting a classification of the audio component based on simple features, followed by a real time rule-based categorization and segmentation of the video object. In [212], the authors use multiple hand-crafted audio features in conjunction with human detection responses and motion information, for violent scene recognition, with optimal performance being obtained by a audiovisual fusion scheme, for both classification and detection tasks.

The authors in [664] use an LSTM network with low-level, tracking and motion visual features combined with two audio information channels in an early fusion manner. The first extracts linguistic information based on MFCC-based ASR text from SEMAINE [559], with the second using a set of 1941 low-level audio descriptors. Experiments on detection on the SEMAINE dataset [425] show that using audio inputs works best – with visual data contributing in some cases – and that LSTM and biLSTM models outperform other classification models examined. In addition, a feature selection post-processing step applied introduces gains only on audio-only inputs.

Given the rich multimodal nature of videos, several approaches utilize multiple modalities, followed by a fusion / aggregation scheme to combine all information into a single video prediction. Multimodal approaches can be seen as a special case of multi-view learning [605, 673], where each modality composes a distinct view of the multi-modal object.

In [526] the authors combine low-level visual motion with MFCC feature responses, applied for TV program genre classification. They report optimal results with a weighted average combination (assigning a 0.7 bias to the audio) with respect to ROC performance. In [179], global color features, audio statistics and motion information of segmented objects and the camera are combined for video genre classification. A similar approach is examined in [621] for visual and temporal modalities. Snoek et al. [586] investigates fusion schemes for visual, audio and textual modalities in the video concept detection task. Specifically, early and late fusion strategies are investigated, aggregating information in the feature level and semantic (prediction) level, respectively. Using SVM classifiers, they report late fusion giving improved average precision scores, at the cost of additional learning effort.

In [579] the authors apply a two-stream DCNN architecture to separately process visual and temporal context (captured by optical flow images [650]), mimicking the human visual ventral and dorsal optical pathways [220]. Experiments on UCF-101 [591] and HMDB-51 [325] datasets show the temporal network (using optical flow) outperforming the spatial network, in terms of accuracy by a 10% absolute score. However, there is significant complementarity between the spatial and the temporal networks, best achieved by a meta-learning process of SVM fusion on softmax scores.

In [688], the authors also use a two-stream spatiotemporal approach like in [579]. They use

two DCNN architectures for deep feature generation (CNN_M [579] and VGG_19 [580]) over two fully-connected layers. Additionally, they examine NN softmax classifications versus a meta-learning phase with linear SVM classifiers. Experiments on the UCF-101 and CCV [286] datasets in terms of mAP show that the deeper VGG_19 network performs better, if large amounts of training data are available. Model (average) fusion (different architectures on the visual stream) performs poorly when fusing networks with different performance, and spatio-temporal linear combination fusion (same architecture, different modality / stream) works well for both datasets, with the spatial part having the larger contribution. In addition, the authors conclude that softmax fares better than SVM meta-learning.

Wu et al. [668, 285] utilize video frames, optical flow images and audio spectrograms, each modal stream fed first to deep convolutional networks and then to an LSTM network. Modality predictions are fused with a set of methods, within which an adaptive approach is proposed which uses class relationships as a regularization mechanism. Experiments on UCF-101 and CCV datasets verify multimodal and CNN / LSTM complementarity and show the proposed fusion outperforming other aggregation techniques. Compared to multiple recent studies, the authors surpass the state of the art in terms of classification accuracy. In [432], the authors compare a variety of learnable temporal pooling approaches along with a two-stream audiovisual DNN Model, proposing a context gating aggregation approach that outperforms competition in the recent Youtube-8M Large-Scale Video Understanding challenge [4].

3.6.2.3 Contributions

Given the existing literature, this study makes the following contributions. First, a comparison of approaches for video classification that vary in architecture complexity and sensitivity to temporal interdependencies is conducted. We seek to investigate the performance of temporally sensitive models versus simple aggregation-based approaches for the video classification task, i.e. the effect of directly modelling temporal inter-dependencies into internal representations versus applying simple aggregation mechanisms. Second, we explore whether relying on the audio or the visual modalities is sufficient for video classification, comparing performance achievable with each modality alone for each approach. Third, we move on to investigate multiple strategies for multimodal fusion that combine these two modalities in a variety of ways, ranging from straightforward solutions to approaches inspired from different machine learning tasks (i.e. image description). We perform both of these comparisons for both types of models. Fourth, we perform an experimental evaluation over multiple datasets, that exhibit different types of annotation (e.g. scene, activity, object, event classes), video content (i.e. visual-centric and audio-centric contents), noise, etc. By using such variable / generic benchmarks for evaluation, we strive to establish a general performance baseline one can expect to reach by using multi-modal features with the proposed classification models, rather than producing the highest possible performance via rigorous fine-tuning and excessive computational cost, for each dataset and task. In the next section, we outline the proposed approaches in detail.

3.6.3 Proposed method

In this section, we describe our proposed method for video classification, which aim to address the following research questions:

1. How do sequential neural models fare versus simpler, aggregation-based approaches that do not consider temporal input inter-dependencies?
2. What is the contribution of the visual and audio modalities, for each of the above model types (e.g. aggregation-based and sequential)? How can video modalities be combined to aid classification performance?

Given these goals, we move on to describe data preprocessing and frame encoding in section 3.6.3.1, followed by a presentation of the single-modality and multimodal workflows examined (sections 3.6.3.2 and 3.6.3.3).

3.6.3.1 Data preprocessing and frame encoding

In this section we outline the preprocessing and encoding steps undertaken to prepare video data to be fed to our classification pipeline. Firstly, we describe the preprocessing stage for the visual content of the video, followed by the spectrogram extraction approach on the audio content and the vector mapping of visual and audio frames produced.

For visual content extraction, we sample video frames at one frame per second, a rate empirically chosen as an adequate trade-off for efficient classification and reduced information load. We subsequently group the frame collection into K video *clips* $\{c_1, c_2, \dots, c_K\}$, each composed of N consecutive frames: $c_i = \{f_1, f_2, \dots, f_N\}$, preserving frame order. For each video, we extract $N_c = 4$ clips, each consisting of $N_{cf} = 8$ frames, amounting to 32 frames per video. Sampled clips are non overlapping and randomly selected from the video frame collection. If the total frames in the video are not enough to support a clip, we duplicate the first frame of the video to reach N_{cf} number of frames, and randomly duplicate clips from those extracted to arrive at N_c clips. A visualization of the visual content preprocessing is depicted in figure 3.24.

We extract audio content from each video and partition it to a sequence of 1-second segments. For each segment, we apply short-time Fourier transform [509] on 20ms windows with a 10ms overlapping stride, producing 99 spectral responses. We keep the 100 most informative coefficients for each response via low-pass filtering, arriving at a 99×100 coefficient matrix, that is stored in an image format as a visual representation of the audio modality. This results in an audio frame sequence per video, that is preprocessed in the same way as the visual modality (see section 3.6.3.1). A visualization of the audio processing pipeline is available in figure 3.25.

Having obtained image frames for the visual and audio modality, them using a DCNN model widely used in image classification tasks, namely the Alexnet architecture [324]; the

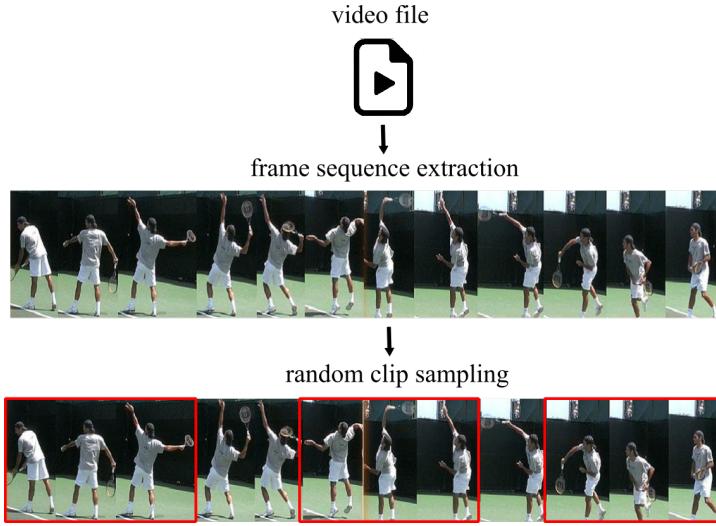


Figure 3.24: The visual content preprocessing pipeline. The underlying frame sequence is extracted from the video from which random clips – i.e. contiguous frame sequences of constant length – are selected. In the example, red rectangles denote the clip boundaries, which are composed of the 3 enclosed frames.

model feeds $224 \times 224 \times 3$ inputs to a pipeline of five convolutional layers with ReLU nonlinearities, max-pooling operators, local response normalization, and three fully-connected (dense) layers. We take advantage of transfer learning approaches [479] and initialize the encoding with pre-trained weights on visual data of the ILSVRC 2012 challenge [535].

The encoding model is used to map input frames to feature vectors; this is achieved by truncating the network before the final prediction softmax layer and obtaining layer responses as the generated feature vector of the input frame. After discarding the task-dependent last dense layer (fc8), we examine truncating up to one of the remaining dense layers in the network, i.e. fc6, fc7, that yield 4096-dimensional activation outputs, which will act as the vectorial representation of input frames.

3.6.3.2 Single-modality workflows

In this section, we outline the components we use for the single-modality classification approaches. This includes the two neural workflows: the aggregation-based FC workflow (section 3.6.3.2) and the temporally-aware LSTM workflow. (section 3.6.3.2).

Additionally, for each workflow we examine a set of fusion strategies that aggregate frame-level to clip / video-level ones.

The FC workflow Given the frame representation mechanism, we move on to produce classification scores for frames, clips and videos. In the FC workflow, we use a “fully-connected classifier”, i.e. a dense layer with $|C|$ neurons that produces $|C|$ -dimensional

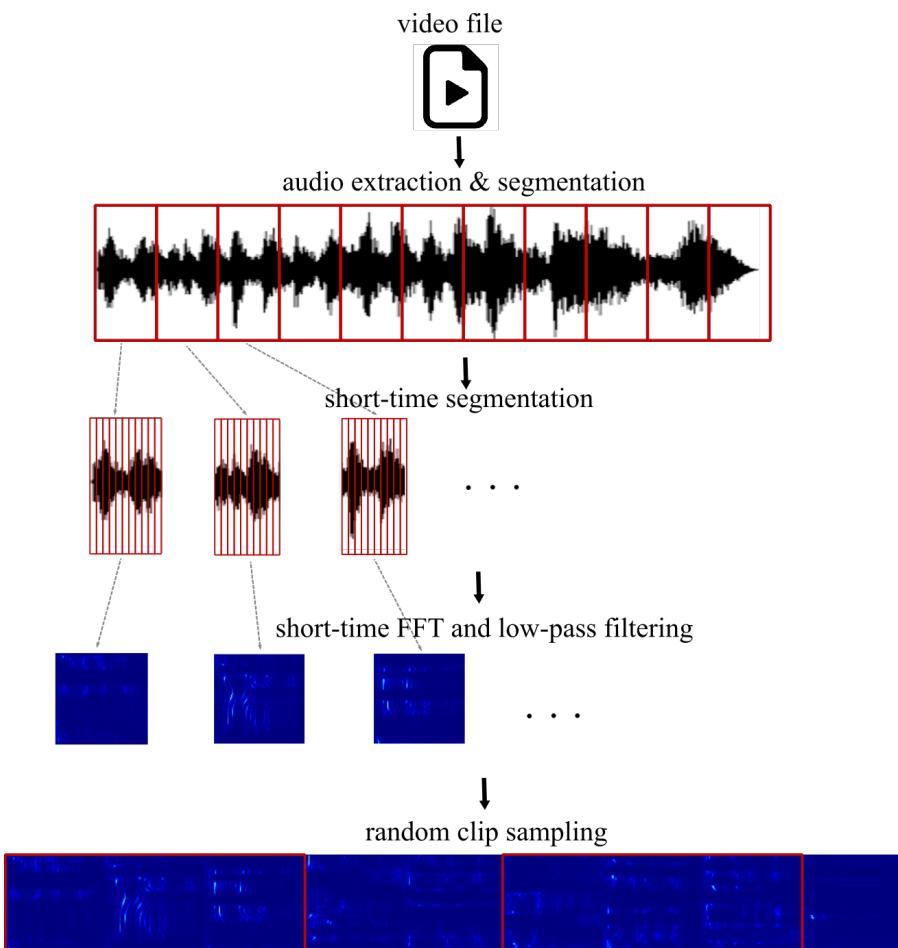


Figure 3.25: The audio content preprocessing pipeline. The audio track is extracted from the video and is partitioned into 1-second segments. Each such segment is mapped into a spectrogram image, via a further 20ms temporal partitioning, followed by application of short-time FFT and low-pass filtering. After the spectrogram sequence is produced, clip extraction is performed in the same way as in the visual content, showcased in figure 3.24.

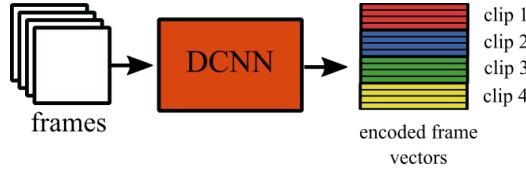


Figure 3.26: The frame encoding process: a collection of video frames (left) is fed through the Alexnet DCNN model (middle). The responses of a selected dense layer are used as frame encodings (right). In this image, there is a single video consisting of 4 clips (color-coded in red, blue, green and yellow), each consisting of 4 frames.

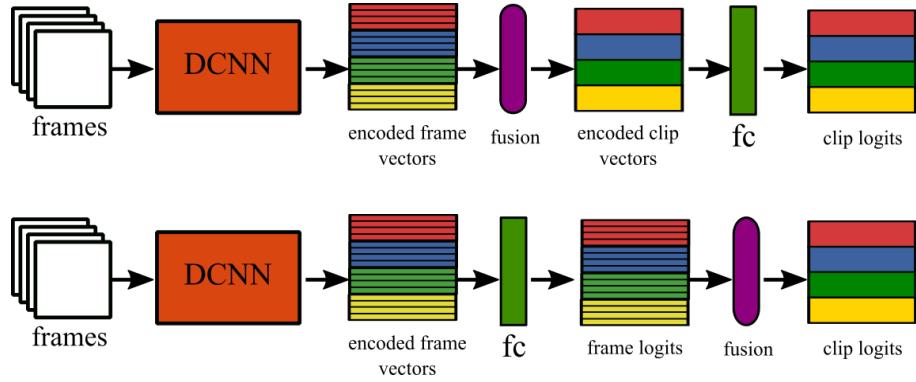


Figure 3.27: Early (top) and late (bottom) frame fusion approaches for the FC workflow. Early fusion acts on encoded frame vectors, fusing frame encodings into a single clip encoding vector fed to the classifier. Late fusion classifies each frame encoding individually, fusing frame-level predictions into a single clip prediction. Color-coding follows the conventions in figure 3.26.

scores, where C is the number of candidate labels. This is followed by a softmax operator that rescales results into a probability distribution. The classifying component is appended to the pretrained network and randomly initialized for training.

Given the encoded frames $\{e_1, e_2, \dots, e_N\}$ from a video, the FC workflow maps e_i to predictions $p_i \in R^{|C|}$, with $0 \leq p_{ij} \leq 1$ denoting the model's confidence that the i -th frame belongs to the j -th class. Since the task of interest is classifying entire videos, we aggregate frame-level predictions into clip-level scores by investigating two fusion approaches. Firstly, early fusion computes the arithmetic mean of frame representations, aggregating all encodings to a single encoded clip vector, which can be classified directly by the FC model to obtain a clip-level prediction. Conversely, late fusion first produces prediction scores for each frame encoding separately and then averages frame scores into one clip-level prediction. The early and late fusion methods are depicted in figure 3.27.

The LSTM workflow The LSTM workflow utilizes a Long Short-Term Memory (LSTM) model [252] to apply sequence-aware analysis on for video classification. Here, frame encodings $\{e_1, e_2, \dots, e_N\}$ are mapped to sequences of responses and hidden states. In

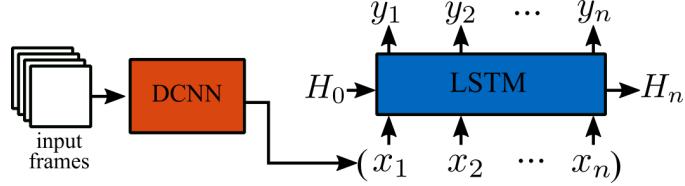


Figure 3.28: The LSTM workflow. DCNN-encoded input frames x_i are fed to the LSTM, which produces outputs y_i . H_0 and H_n denote the initial and final state vectors.

the response sequence $Y = \{y_1, y_2 \dots, y_N\}$, y_i is a vector in $\mathbb{R}^{|C|}$, with y_{ij} representing the model's confidence that the i -th frame belongs to the j -th class, given the input e_i and the current hidden state h_i . Each step updates the hidden state of the model (i.e. the state in step i holds information from steps $j < i$), capturing temporal interdependencies in the sequence and making the LSTM workflow sequence-aware. Evidently, frame k has a cumulative contribution on the network, affecting all subsequent hidden states and outputs. The workflow is illustrated in figure 3.28.

In order to arrive at clip-level predictions, we experiment with selecting the prediction at the last time step (*last* strategy), i.e. utilizing the built-in mechanism (via updating hidden state) for temporal aggregation in the LSTM. Additionally, we consider taking the average of all prediction steps (*avg* strategy) and treating the hidden state as the final prediction vector (*state* strategy). The latter method treats the LSTM as a prediction encoder, representing the input sequence as a model vector in its internal state. This is accomplished by setting the number of state neurons to the number of desired classes and considering the final state vector h_N as the classification response for the input frame sequence.

3.6.3.3 Multimodal workflows

In this section, we describe multimodal classification workflows. Two approaches are examined : “direct” multimodal fusion approaches (section 3.6.3.3) use straightforward techniques of combining data from different modalities, while “bias fusion” (section 3.6.3.3) considers approaches inspired image description, where additional modality information is presented as a bias in the input sequence.

Direct data fusion In this section, we propose fusion approaches similar to the work of [688], but applied on audio-visual content rather than spatial and temporal streams. We examine two methods for data combination, given visual and audio data. First, *avg* fusion combines modality data by computing the arithmetic mean at the frame encoding level, i.e. after images have been mapped to a vector representation (see section 3.6.3.1). Specifically, given visual and audio input clips $c_v = \{e_{v1}, e_{v2}, \dots, e_{vN}\}$, $c_a = \{e_{a1}, e_{a2}, \dots, e_{aN}\}$, *avg* fusion produces the multimodal clip $c_m = \{e_{m1}, e_{m2}, \dots, e_{mN}\}$, where $e_{mi} = \frac{e_{vi} + e_{ai}}{2}$. Similarly, *max* fusion computes coordinate-wise maxima from two clips, i.e. $e_{mi} = \max(e_{vi}, e_{ai})$. Furthermore, *concat* fusion concatenates the vectors repre-

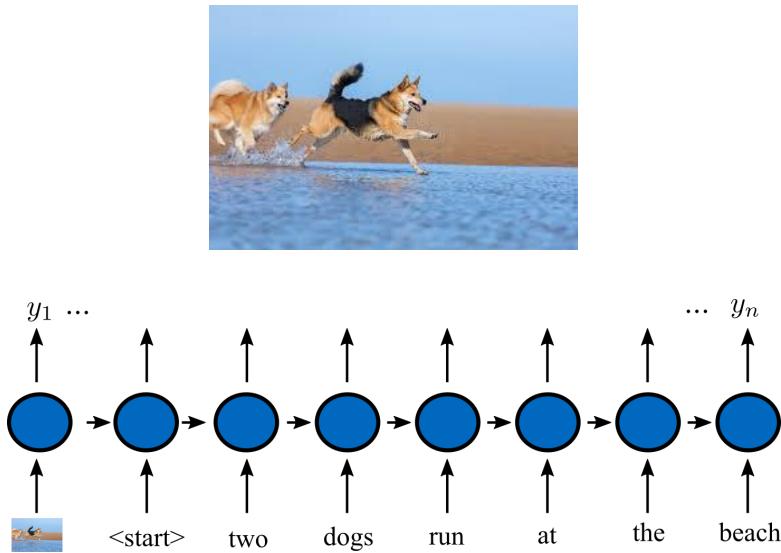


Figure 3.29: Example of handling an image description task with a recurrent model. Given the input image (top), information pertaining to it is supplied in the first input position in the recurrent model (bottom). It is followed by caption information tokenized to words. The `<start>` item is a special token that denotes the beginning of the caption word sequence.

sentations from each modality, i.e. produces clip with vectors $e_{mi} = [e_{vi}^T, e_{ai}^T]^T$, where x^T is the transpose of x . After direct data fusion, the multimodal clip c_m can be classified by the FC or LSTM workflow (see section 3.6.3.2) without any modification required.

Sequence bias fusion In this section, we investigate fusion methods inspired from neural sequence models for tasks such as image description task, where given an input image, the goal is to produce a caption that best describes the visual content. Given the sequential nature of a text caption, the use of recurrent deep neural models is an established approach [154, 643, 302, 398].

The first method, *input-bias*, is inspired from the approach in [643]. There, images and caption words are encoded into vectors with the visual vector fed as the input for the first time step, as a way to introduce bias to the sequence that follows. Given the significant effect of the first frame in the sequence (see section 3.6.3.2) information from this step propagates to influence the model to correctly interpret the word sequence that follows. The mechanism is depicted in figure 3.29.

We adopt a similar logic for multimodal fusion: we label the visual contents the “main”, and the audio contents the “auxiliary” modality in the video. The auxiliary modality is aggregated into a single clip vector via average fusion and introduced as bias (like the visual vector in image description), while the main modality retains its sequential structure (analogous to the caption words in image description) and is fed to the sequential model after the bias. Figures 3.30, 3.31 provide a depiction of the procedure. The process can be viewed as imbuing “main” frames with an additional, special frame at the start.

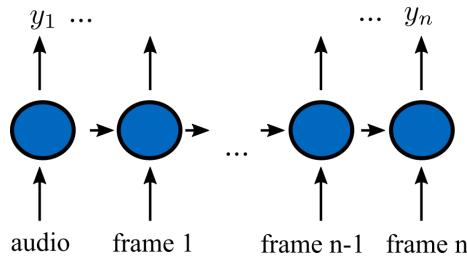


Figure 3.30: The input-bias method for introducing audio context in the visual frame sequence. Aggregated audio information (the “auxiliary” channel) is supplied as the first input element, followed by the visual encoded frames of the “main” modality.

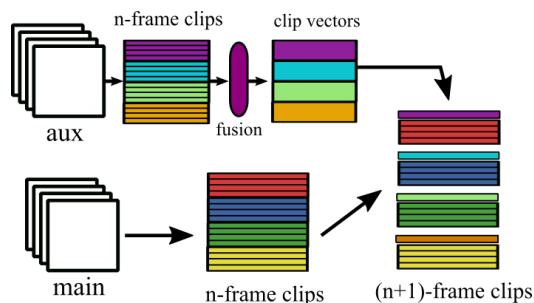


Figure 3.31: The input-bias multimodal fusion method. Frame vector data from the auxiliary modality (top diagram) are fused into clip vectors and inserted as the first vector in the corresponding main modality sequence.

Obviously, the audio and visual frame encodings need to be mapped to a vector space of same dimensionality.

The second method modifies the bias mechanism by feeding the bias vector as the initial state for an LSTM model, i.e. a *state-bias*. This initialization aims to condition the internal state of the network directly, rather than providing an introductory input step and letting the network produce an appropriate first state (after the consumption of the bias, i.e. h_1) on its own. The approach assumes that bias information will hold patterns that the network will be able to interpret as a meaningful initial hidden state seed. Secondly, since this method is applicable only to RNN-like models, we only apply it only in the LSTM workflow. A visualization is depicted in figure 3.32.

Late video-level fusion While the previous multimodal fusion methods combine frame encodings from different modalities and subsequently apply a classification process with workflows from sections 3.6.3.2 and 3.6.3.2, *video-level late fusion* directly aggregates classification results. First, we compute a linear combination of the marginal modality scores, setting complementary visual and audio modality weights, i.e. $w \in [0, 1]$ and $1 - w$, varying w with a step of 0.1. See figure 3.33 for a visualization of the method. Secondly, we examine max pooling aggregation, selecting the best performing single-modality score for the video.

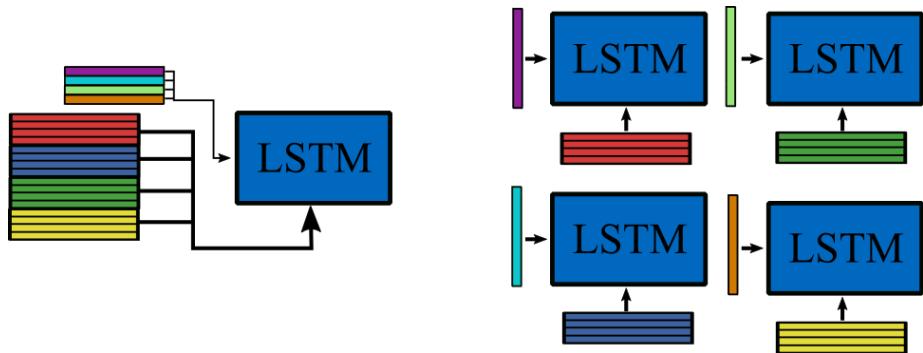


Figure 3.32: The state-bias multimodal fusion method. On the left, the input sequences are fed to the LSTM input and the state vectors are set as the initial LSTM state. In the image to the right, the same process is shown at a per-clip basis. Encoding vector colors denote which clips the vectors belong to.

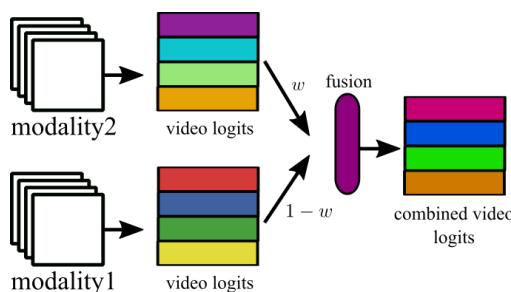


Figure 3.33: Multimodal late fusion.

In this study, the marginal visual and audio modalities selected to be combined with *late-video* fusion, are the best-performing single-modality workflow for each modality. This scheme is simple but its practical application has some disadvantages. First, picking the best performing workflow imposes an implied prior selection procedure. While we include such a process as a part of our single-modality investigation, its execution in a generic multimodal classification setting is impractical and cumbersome. Secondly, the late, video-level combination of predictions of two different models requires two different and separate training runs to produce the distinct visual and audio models, rather than handling multimodal content in an end-to-end multimodal setting directly, as is the case with the other methods examined.

3.6.4 Experimental results

This section describes the experimental evaluation of our proposed method for the task of multimodal video classification. We first present the datasets utilized and the experimental setup adopted in section 3.6.4.1. We then investigate the performance of proposed approaches on the single-modality and multimodal settings in sections 3.6.4.2 and 3.6.4.3, respectively. Finally, we discuss the obtained results in section 3.6.4.2.

3.6.4.1 Datasets and Experimental Setup

We use a number of datasets to evaluate our methods, presented below.

1. The UCF-101 [591] human action recognition dataset consists of 13320 YouTube⁵⁵ videos spanning 101 categories which can be grouped in 5 broad supersets, namely Human-Object Interaction, Body-Motion Only, Human-Human Interaction, Playing Musical Instruments and Sports. Video quality varies, both in terms of visual and audio content quality. We divide the videos into the train / test sets using the respective partitions used in [154], resulting in 9537 and 3783 train / test videos, respectively. Since our multimodal workflows utilize audio, we discard classes for which no videos with audio content exist. This results in 51 classes and 6837 videos with audio, corresponding to 4839 and 1944 train / test videos. We call this modified dataset UCF-51.
2. The KTH dataset [560] is a human actions dataset, consisting of 6 classes. It consists of 599 videos, split to 389 and 210 train / test videos. In KTH, videos are simple, with no occlusion and with the actor centered in simple scenes. Videos lack both color and audio, so we use this dataset only for the first stated goal of this study (i.e. handling of the temporal content) and not for the audiovisual multimodal workflows.
3. Audioset [194] consists of videos from YouTube, forming a hierarchical ontology of audio events of 632 classes. Since we deal with single-label classification, we re-

⁵⁵<https://www.youtube.com>

Table 3.15: The datasets used in the experimental evaluation.

dataset	classes	video instances		min / max samples per class		notes
		train	test	train	test	
UCF-51 [591]	51	4893	1944	76 / 120	28 / 48	human activity classes poor / irrelevant audio
UCF-10 [591]	10	783	314	76 / 83	28 / 34	human activity classes poor / irrelevant audio
KTH [560]	6	389	210	64 / 65	35 / 35	human activity classes no color or audio
AudioSet [194]	43	2602	2618	8 / 226	19 / 170	audio-centric classes imbalanced
CCV [286]	20	2708	2759	41 / 287	43 / 295	diverse classes

stricted the video classes to leafs in the ontology tree. Additionally, we keep classes that are annotated with a high quality, with respect to the provided class-wise annotation quality index ⁵⁶. Specifically, we kept a quality value of 1.0, resulting in 43 retained classes. We downloaded the respective videos from YouTube via the provided urls with the youtube-dl tool ⁵⁷. Not all listed videos were available, resulting in 2602 train and 2618 test videos, with a notable imbalance among the class samples.

4. The Columbia Consumer Video (CCV) dataset [286] consists of user videos from YouTube consisting of 20 diverse classes (e.g. human activities, scenes, objects) and providing pre-trained hand-crafted audiovisual features (SIFT, STIP, MFCC features). Videos are provided via a list of YouTube video identifiers, which we downloaded with the youtube-dl tool. The retrievable and available videos amounted to 2708 train and 2759 test instances.

Detailed results on each dataset are presented in table 3.15. The datasets collected are diverse in terms of video classification task (e.g. action, event, scene, object recognition), number of classes, video quality (e.g. image / sound artifacts and noise, audio relevance to the ground truth class), classification difficulty (e.g. per-class variability in, e.g., video angle, pose, actors, environment and background) and completeness (e.g. manually downloaded and partially available data versus a complete list of provided data).

We used python3 and tensorflow 1.4 ⁵⁸ [2] to construct a complete video classification suite, including dataset preprocessing, serialization, classification, large-scale experiment execution and monitoring tools.

The code is available on GitHub ⁵⁹. We used the “Caffe reference” implementation ⁶⁰ for the DCNN Alexnet model. We used tensorflow’s TFRecord format for data serialization and the ffmpeg utility to extract visual frames and mp3 audio files from a given video. Regard-

⁵⁶<https://research.google.com/audioset/dataset/index.html>

⁵⁷<https://rg3.github.io/youtube-dl/>

⁵⁸<https://www.tensorflow.org>

⁵⁹<https://github.com/npit/video-learning-tf>

⁶⁰http://caffe.berkeleyvision.org/model_zoo.html

ing spectrogram extraction, we used the `pyAudioAnalysis`⁶¹ [211] library. Experiments ran on an Ubuntu 16.04 system, with a Tesla K40 GPU.

We conducted a set of preliminary experiments to tune the hyper-parameters for each proposed approach via grid search, as well as limit the number of configurations required to be subjected to full evaluation. For the FC workflow, we vary the fully-connected layer of AlexNet that is utilized for frame encoding (*fc6* or *fc7*) and the frame fusion method (i.e. early or late fusion) for generation of clip-level results. For the LSTM workflow, we vary the fully-connected layer, the number of layers in the LSTM (evaluating up to 5 layers), the number of neurons in each hidden layer (ranging from 200 to 2000) and how the final LSTM representation is obtained (i.e. *avg*, *last* or *max* pooling if sequence step outputs). The above search favored using the *fc7* layer with late fusion for the FC workflow and the *fc6* layer on a 2-layer LSTM with 500-neuron hidden states with *avg* fusion, for the LSTM workflow.

Training is performed with the cross-entropy loss, mini-batch stochastic gradient descent [72] on a batch size of 20 videos (640 frames) and a learning rate of 0.1, exponentially decreasing 100 times on the course of training. We train with an early stopping criterion where the model with the best test performance is retained within an overall training process up to 10 epochs. We evaluate classification with the accuracy measure.

3.6.4.2 Single-modality experiments

In this section we describe the experiments evaluating our proposed workflows on the single-modality setting, in an attempt to address the first goal of this study, as described in section 3.6.2.3. To recall, we seek to evaluate sequential and aggregation-based feed-forward neural models, on their ability to utilize the temporal component in a video and the extent to which this contribution has an effect on video classification. In this study, the goal is tackled by comparing the LSTM and FC workflows, which represent the two aforementioned model categories, in the video classification task.

Results In order to evaluate each proposed workflow, we set up video classification experiments for the datasets outlined in 3.6.4.1. We extract the visual and audio modality data and apply the FC and LSTM workflow on each one. Each model is trained with the same parameters for training, clip and frame extraction as in the preliminary experiments, which, under an empirical evaluation, were found adequate for the experiments in this section as well. For each dataset examined, we compute the chance and majority classifier baselines. The first selects one of the available classes at random, while the second selects the majority class in the dataset samples. Workflow and baseline results are reported in table 3.16. The column “diff %” refers to the relative performance change from the FC to the LSTM workflow.

For KTH classification, despite ignoring frame inter-dependencies in the video sequence,

⁶¹<https://github.com/tyiannak/pyAudioAnalysis>

Table 3.16: Single-modality collective results for all datasets. Bold values indicate dataset-wise maxima, while underlines indicate modality-wise maxima.

dataset	visual			audio			baseline	
	FC	LSTM	diff %	FC	LSTM	diff %	chance	majority
KTH	0.814	0.7523	-8.2	N/A	N/A	-	0.1667	0.1667
UCF-51	0.5889	0.6604	10.8	0.2788	0.2654	-5.0	0.0196	0.025
AudioSet	0.267	0.309	13.8	0.340	0.276	-23.2	0.023	0.064
CCV	0.653	0.657	0.6	<u>0.367</u>	0.342	-7.4	0.05	0.106

the FC workflow outperforms the LSTM one by a relative 8.2% increase. This may be due to the visual content in KTH videos being simple and lacking variation, as stated in 3.6.4.1, rendering the content simple enough for the fully-connected classifier to handle, and at the same time, for the LSTM classifier to overfit. Finally, the worst-performing proposed method introduces a four-fold increase on the highest baseline accuracy.

Regarding the UCF-51 dataset, visual content runs outperform audio content runs for both workflows, by more than double accuracy scores. The FC workflow outperforms the LSTM workflow by 5.2% with respect to audio content, while the LSTM workflow outperforms the FC workflow with respect to visual content by a factor of 12%. The worst-performing proposed method achieves an approximately ten-fold increase on the naive classification baselines.

For AudioSet, the FC workflow on audio data emerges as the best performer in the dataset, with an accuracy score of 34.1%, a 27.7% better score than the second best performer, i.e. the LSTM applied on visual video content with 31%. We thus get no definitive results for this dataset, with no modality and workflow being consistently better than their competition. Regarding audio content, the FC workflow outperforms the LSTM workflow by a significant relative factor of 23.1%. For the visual modality the LSTM workflow fares better than the FC workflow by 16.1%. The worst-performing method increases the baseline performance by a factor of three.

For the CCV dataset, the visual modality outperforms the audio modality, for all workflows and hand-crafted features. For the proposed workflows, the relative performance difference is 77.7% and 92.1%, for the FC and LSTM workflows, respectively. The visual LSTM is the best performer, very closely followed by the visual FC with a 0.6% relative performance difference. Regarding audio, the FC workflow outperforms LSTM by a factor of about 7.6%. We additionally report performance of pretrained, video-level visual and audio features which are provided with the dataset, namely handcrafted visual descriptors (SIFT [395], STIP [341]) and the MFCC descriptor [187] for audio, which result in 0.491, 0.39 and 0.304 accuracy scores, respectively. Results show that regarding hand-crafted features, the established SIFT descriptor outperforms STIP features by a factor of 25.5%. The proposed workflows outperform the handcrafted features, by a considerable margin (with respective min / max relative improvements of 12.5% and 68%). The worst performing proposed approaches introduces a more than double improvement on baseline classification.

Discussion Here we attempt to summarize the findings of the single-modality experiments and connect them to the goals of this study. The experimental results of the proposed methods are collectively illustrated in tables 3.16. Applied on the proposed methods of the study, the experiments described in this section can provide evidence on the suitability of the feedforward FC workflow versus the recurrent LSTM workflow that considers input inter-dependencies. In addition, the aforementioned suitability is investigated in a multimodal setting, i.e. for the visual and audio modalities, all with respect to the video classification task. Thus, in light of the experimental results for each dataset, we can arrive at the following conclusions:

1. The visual modality outperforms the audio modality everywhere but Audioset. This can be explained by the ontology and content of the latter, where both the videos and event classes are audio-related. In addition, the dataset was filtered to high-quality samples, limiting the occurrence of videos with unrelated audio (e.g. music, narration, overlaid audio effects, as explained in section 3.6.4.1 which further enhances the significance of audio and reduces its potential to introduce harmful noise. It should be noted, however, that the performance difference between modalities in these cases is nowhere near similar. For the cases where the visual modality outperforms the audio one, it does so with a mean relative accuracy difference of approximately 130% and 85% in UCF-51 and CCV, respectively. However, the audio modality outperforms the visual one by approximately 7%, for Audioset. This hints at the visual modality being an important discriminatory information channel in video classification, as well as hinting towards the primacy of the visual modality in human perception, categorization and semantic segmentation.
2. The pre-trained weights of the frame encoding Alexnet DCNN manage to provide a good initialization point not only for the visual modality, but for spectrogram image encoding as well. The latter can be illustrated by performance in Audioset, where the audio modality runs outperform the visual one. Had the representation been inadequate to capture spectrogram information, the audio modality runs would behave consistently poorly in all datasets.
3. We can see that the LSTM workflow outperforms the FC workflow on visual content for all datasets we examined, except in KTH. As expressed in the previous section, this may be attributable to the wealth of motion-related information in the visual modality, which can be effectively captured by the LSTM model – the simplicity of KTH data causing the LSTM model to overfit on irrelevant video features. However, the FC workflow in turn outperforms the LSTM workflow on audio content, for all datasets examined. A possible explanation for this is that temporal input dependencies in audio spectrogram images are qualitatively different from visual motion cues, and they cannot be efficiently captured by the LSTM classifier with the current configuration. Another possible culprit could be the Alexnet vector encoding being inadequate for capturing spectrogram temporal inter-dependencies. Since the encoding approach yields good results with the FC workflow however, a more probable avenue for the efficient application of the LSTM workflow could be a modification of

the training parameters or the classifier’s architecture itself, so as to bring about a better fit of the model on the encoded spectrogram input sequences.

4. The handcrafted features examined are outperformed by the proposed FC and LSTM workflows in both modalities. This verifies the superior expressive capabilities of DNN-based deep distributed features. This observation concerns the CCV dataset, which is the only dataset among the ones examined that provided handcrafted audiovisual features.

3.6.4.3 Multimodal experiments

In this section we tackle the second goal of this study (see 3.6.2.3), i.e. an examination of various multimodal approaches for the video classification task. While in the previous section we examined the proposed workflows for each modality separately, here we apply the proposed audiovisual fusion methods described in section 3.6.3.3. As in the previous section, we use data from the datasets described in section 3.6.4.1. We exclude the KTH dataset due to its lack of audio content. In section 3.6.4.3 we describe the experimental setting and results per dataset, followed by a summary of the results in section 3.6.4.3.

Results In the sections below we present the experimental results of the multimodal workflows, summarized in table 3.17. For the *late-video* mutlimodal approach, we vary the visual modality weight w at a increment of 0.1, with the audio modality having a corresponding weight of $1 - w$, and report the best performance and weight.

Regarding UCF-51, LSTM outperforms FC in all shared fusion methods (i.e. *avg*, *ibias*, *max* and *concat*). All LSTM fusion methods manage to outperform single-modality baselines, except *sbias* fusion. On the other hand, FC fusion can not exceed the best single-modality visual run (with a relative 3.78% lower accuracy). All multimodal fusion methods outperform the best audio single-modality run. For FC fusion, the *avg* method performs best, followed by *concat*, *ibias* and *max*, with each method having relative neighbouring performance differences of 2.75%, 1.95% and 2.71% respectively. Regarding LSTM fusion, the *concat* method appears to fare best, followed by *avg*, *max*, *ibias* and *sbias* aggregation. Their relative performance difference is 0.84%, 1.43%, 3.72% and 3.22%, in the aforementioned order, respectively. For *late-video* fusion, we acquire a best linear combination performance of 0.72119 with visual weight 0.8, while maximum fusion performs significantly worse. The large visual weight in the combination is not surprising, given the performance difference of the visual modality over the audio modality noted for UCF-51 in section 3.6.4.2. Finally, *late-video* fusion outperforms FC fusion, LSTM fusion and single-modality runs, achieving relative increase of 13.5% and 1.12% over the next best approaches in FC and LSTM fusion.

Regarding Audioset, the FC workflow outperforms LSTM in all shared fusion methods. All FC fusion methods outperform single-modality baselines, while LSTM fusion exceeds best single-modality performance only with the *input – bias* fusion method. For FC fu-

Table 3.17: Multimodal fusion collective results for all datasets. Regarding the multimodal workflow-based runs (i.e. FC or LSTM runs, excluding *late-video* fusion), bold values indicate dataset-wise maxima, while underlined ones represent workflow-wise maximum values, for the given dataset. For *late-video* runs, *LC* denotes linear combination *late-video* fusion runs, with the optimal weight included in parentheses. In addition, *max* stands for maximum *late-video* fusion and we highlight *late-video* maximum values with italics.

dataset	FC				LSTM					<i>late-video</i>	
	<i>avg</i>	<i>concat</i>	<i>ibias</i>	<i>max</i>	<i>avg</i>	<i>concat</i>	<i>ibias</i>	<i>max</i>	<i>sbias</i>	<i>LC</i> (w)	<i>max</i>
UCF-51	<u>0.635</u>	0.618	0.606	0.590	0.707	0.713	0.672	0.697	0.65	<u>0.721</u> (0.8)	0.376
AudioSet	0.388	0.378	0.384	0.192	0.248	0.28	<u>0.367</u>	0.225	0.258	<u>0.464</u> (0.6)	0.351
CCV	0.643	0.635	0.639	0.634	0.673	0.679	0.679	0.67	0.639	0.693 (0.6)	0.655

sion, the *avg* method performs best, followed by *ibias*, *concat* and at a very low accuracy, *max*. Relative neighbouring performance differences of the methods lie at 1.04%, 1.58% and 96.8% respectively. Regarding LSTM fusion, the best method is *ibias*, followed by the *concat*, *sbias*, *avg* and *max* methods. Relative performance differences are 31.07%, 8.52%, 4.03% and 10.2%, in the aforementioned order, respectively. Regarding *late-video* fusion, an optimal combination performance of 0.46409 is obtained with a visual weight 0.6. It is noteworthy that despite the audio modality outperforming the visual modality for AudioSet, the best combination results are obtained with a larger weight for the visual component. Maximum *late-video* fusion performs close above the single-modality scores, at an accuracy of 0.351. Finally, *late-video* fusion outperforms FC fusion, LSTM fusion and single-modality runs. The performance increase over the best proposed multimodal fusion runs are 19.58% and 26.34%, respectively.

Finally, with respect to CCV, the FC workflow outperforms LSTM in all shared fusion methods. LSTM fusion generally outperforms the single-modality visual baseline, except from *sbias* aggregation. No FC fusion method manages the previous, however. Both workflows exceed the best audio single-modality run with all aggregation methods. LSTM fusion outperforms the FC approach for all shared fusion methods. For FC fusion, the *avg* method performs best, followed by *ibias*, *concat* and *max*. The relative performance difference between the aggregation methods is 0.62% for the first two pairs, and 0.1% for the last. Regarding LSTM fusion, the best methods are *ibias* and *concat*, followed by the *avg*, *max* and *sbias* aggregation. The relative performance differences are 0.89%, 0.44% and 4.85%. *late-video* fusion produces a best result of 0.693 at a linear combination with a visual weight of 0.6, while maximum fusion performs very close to the visual baseline, at 0.655 accuracy. Additionally, *late-video* fusion outperforms FC fusion, LSTM fusion and single-modality runs, achieving 7.77% and 2.06% relative accuracy increases over the best proposed multimodal fusion runs, respectively.

Discussion At this point we summarize the findings of the multimodal experiments, extending the investigation of the single-modality experiments to modality aggregation approaches for video classification and (multimodal classification in general). With respect to the study goals, we condense the experimental conclusions in order to address the two

Table 3.18: Multimodal fusion method average ranks, with respect to each dataset, the multimodal fusion workflow and overall. Bold values indicate row-wise minima (with respect to the rank reference), while underlined values indicate group and column-wise minima (with respect to the both fusion method and the reference type)

rank reference	fusion methods				
	<i>avg</i>	<i>concat</i>	<i>ibias</i>	<i>max</i>	<i>sbias</i>
datasets					
UCF51	<u>3.0</u>	<u>3.0</u>	5.0	<u>5.0</u>	<u>4.0</u>
CCV	<u>3.0</u>	3.5	3.5	5.5	5.0
Audioset	<u>3.0</u>	<u>3.0</u>	<u>2.0</u>	7.5	5.0
total	<u>3.0</u>	3.167	3.5	6.0	4.667
workflows	<i>avg</i>	<i>concat</i>	<i>ibias</i>	<i>max</i>	<i>sbias</i>
LSTM	<u>3.0</u>	<u>1.333</u>	<u>2.333</u>	<u>4.0</u>	4.667
FC	<u>3.0</u>	5.0	4.667	8.0	N/A
total	<u>3.0</u>	3.15	3.5	6.0	4.667

Table 3.19: Relative percent performance of the LSTM to the FC workflow, per fusion variant and dataset

datasets	<i>avg</i>	<i>concat</i>	<i>ibias</i>	<i>max</i>
UCF51	11.34	15.37	10.89	-18.14
AudioSet	-36.08	-25.93	-4.43	17.19
CCV	4.67	6.93	6.26	5.68

stated questions of the former: first, the comparison of feedforward (i.e. the FC workflow) and recurrent (i.e. the LSTM workflow) deep neural models, with respect to their ability to capture the temporal video components, and secondly, the investigation of the effect of multimodal approaches, for the video classification task.

We make the following observations, given the collected results of the multimodal experiments displayed in table 3.17, the extracted average rank information of the multimodal fusion methods used, in table 3.18.

- The FC fails to surpass single-modality baselines for all datasets except Audioset, which is the only dataset where it outperforms the LSTM workflow. This can be explained by the affinity of the FC workflow for the audio modality and the audio-oriented data of Audioset, as explained in section 3.6.4.2.
- Regarding fusion methods for the FC workflow, the *avg* aggregation method is the top performer on average, followed by *ibias* and *concat*. *max* fusion performs worst, for all datasets.
- The LSTM workflow outperforms the FC workflow, for all datasets but Audioset, as explained above.
- The LSTM workflow surpasses the single-modality best performing runs in every dataset with some aggregation method. However, no aggregation method can be chosen that does so consistently.
- The LSTM workflow performs best, on average, when paired with the *concat* fusion method, which is top performer in 2 out of 3 datasets examined. The rest of the methods in order of performance are *ibias*, *avg*, *max* and finally the *sbias* method.
- Comparing fusion method performance across workflows in table 3.19, it appears that the LSTM workflow generally outperforms the FC workflow for every fusion method but *max*, in every dataset except Audioset. This reinforces the findings of the single-modality experiments, over the LSTM classifier's ability to capture temporal context in the input that contributes to video classification. This conclusion does not hold for Audioset, a possible reason being the characteristics of the dataset, as explained in 3.6.4.2.
- In total, simple frame averaging via *avg* fusion emerges as the best method on average, outperforming concatenation via *concat* fusion. The large feature vector dimensionality of the latter possibly requires a more complex model and / or additional training, the former of which is reflected by the improved performance on the much more expressive LSTM model, when compared to the fc classification of the FC workflow. The sequence bias introduction approach via *ibias* does not seem to provide a better fusion approach, although performing better on the sequence-oriented LSTM workflow. *max* fusion does not produce good results, indicating that marginal modality information should be combined, rather than discarded. Finally, the *sbias* method is the worst performing fusion approach. This can probably be explained by the low dimensionality of the state vector (i.e. set to a number of neurons the

number of desired classes), which, although it enabled the formation of model vector representation, it seemed to prevent the LSTM from retaining sequence-related information effectively.

- Regarding *late-video* fusion, the linear combination approach consistently outperforms the FC and LSTM multimodal workflows. We can identify a preference for an increased bias towards the visual modality (even for AudioSet) with all optimal visual weights exceeding 0.5. However, no unique optimal weight pair can be deduced that works best across all datasets. A naive result can be obtained by averaging the accuracy scores across datasets and collecting the weights of the best aggregate accuracy. Doing this, we acquire a visual / audio weight pair of (0.7, 0.3).

3.6.5 Comparison to other systems

In this section we present a brief comparison to related work on video classification, for the selected datasets. This comparison is not entirely valid and straightforward, since we use reduced class sets for some of these datasets for the reasons outlined in section 3.6.4.1. In addition, our experiments focus on answering the questions stated in 3.6.2.3, rather than aiming to surpass the state of the art for each of the examined datasets and corresponding specific classification task. This results in an overall reduced performance per dataset. Despite these observations, we include the comparative results below, for completeness.

In classification experiments on the KTH dataset, the authors in [342] achieve a performance close to 90% using HoG features in a bagging configuration. More recent approaches adopt CNN features, such as the work in [520], where DCNN spatiotemporal features in a pyramidal hierarchy manage to push the state of the art closer to 95%. For the UCF-101 dataset, a significant performance increase is achievable via the inclusion of temporal features in the multimodal setting and by using deep models for feature generation. Specifically, accuracy scores above 87% can be approached, while multiple model fusion techniques result in a performance around 92% [669, 285]. Very recent approaches achieve an accuracy of 95% [668].

For CCV, we already reported the superiority of the deep learned features on visual and audio content, when compared to the provided visual and audio handcrafted features. Recent optimal results using SIFT features are obtainable by Fisher vector aggregation, which is the approach adopted [453], producing results with an accuracy of 71.7%. Given deep approaches, the same trends of using temporal features (e.g. image flow and motion trajectories) as well as deeper convolutional models than Alexnet are popular in recent works. For example, a classification performance of 84% can be achieved with audio, video and temporal features, using a regularized context-analysis scheme in [285].

Regarding AudioSet, the deep audio features provided with the dataset (extracted with a VGG model in the same way as in [242] and pre-trained on YouTube-8M dataset [4]) reach an accuracy of 46% when evaluated with the FC workflow and a mAP 0.31 as reported in [242]. Other approaches use deep DCNNs with multi-label training, taking advantage of

the class ontology hierarchy of the dataset [326], reaching a mAP of 0.21 or using an attention mechanism in [319, 692], reporting a mAP of 0.327 and 0.36, respectively.

Possible extensions to the work of this study are outlined in the following section (3.6.6.3) and provide a number of ways and generic guidelines of approaching the aforementioned performance improvements.

3.6.6 Conclusions

This section concludes the study by offering a summary of the goals, the proposed models and the conclusions that can be drawn from the experimental results. In section 3.6.6.1, we provide the aforementioned summary, providing a brief description of the problem tackled in the study, the stated goals and the proposed approaches to reach them. In section 3.6.6.2, we layout the main findings extracted from the experimental evaluation of the proposed methods. In section 3.6.6.3 we present a number of ways to extend the investigation performed in this work in various directions, in light of the technical details, method assumptions and acquired results.

3.6.6.1 Summary

In this study, we examined the multimodal video classification task, entailing the automated prediction of video labels relevant to the content present in the video. Given the inherent multimodal nature of the latter, this prediction process was designed to take into account different modalities, namely the visual, audio and temporal video data streams. We handle the audio and visual data directly, by extracting spectrogram and video frame sequences respectively, feeding them to a deep neural classifier. On the other hand, we consider the temporal context indirectly, by examining classification models with varying sensitivity to the temporal inter-dependencies of the input sequence items. Given these aspects of our classification pipeline, we set the following research goals for this study, with respect to video classification:

1. How do feedforward, aggregation-based models perform, compared to sequence-based models that consider temporal inter-dependencies? How does this relationship change for the visual and audio modalities?
2. What is the performance of the aforementioned models in the multimodal, audiovisual setting? How can modality data be combined to improve video classification score?

The first goal was examined by instantiating two video classification workflows representative of the described approaches: the feedforward FC and the sequence-aware LSTM workflow. Each approach was applied on multiple, diverse datasets after a set of preliminary experiments with which architecture meta-parameters were fine-tuned. Regarding

the second goal, we applied these workflows with three general strategies of modality fusion. These combined the visual and audio data representation directly (“direct” data fusion), introduced an audio bias on the visual information sequence (sequence-bias techniques) or applied a late fusion on video-level prediction scores (late-video fusion). Both of these goals can be condensed into a formulation of a multimodal video classification baseline configuration, illustrating generic estimates of expected performance on each dataset, with a deep multimodal classification pipeline applicable to any video classification task.

3.6.6.2 Findings

Given the experimental evaluation results, we arrived at a number of findings per stated goal. For the comparative performance of the feedforward FC workflow and the recurrent LSTM, the latter approach is the more suitable choice for visual data in general, albeit the model can overfit very simple datasets. On the other hand, the FC workflow performs consistently better on audio content when represented by spectrogram images. Furthermore, while the relative significance of the visual and audio modalities for video classification depends on the underlying dataset and corresponding annotation, we found the visual modality to be the significant information channel, significantly outperforming the audio modality in virtually all datasets examined. The exception to this is the audio-inclined Audioset, where the tables are turned but with a far lesser performance difference. Finally, we verified the superiority of DNN-based learned representations with respect to hand-crafted features, for the CCV dataset, where handcrafted features were provided.

Furthermore, we examined audiovisual fusion approaches for multimodal video classification, each examined with the aforementioned network architectures. Despite practical disadvantages, the *late-video* linear combination fusion approach produced the best multimodal fusion results, while, conversely, the max-pooled variant performs much worse, close to single-modality baselines. With respect to the other approaches, we identified the suitability of the *avg* and *concat* fusion methods for the FC and LSTM workflows respectively and in general, among all fusion methods on average. Max-pooling modality fusion performed poorly here as well as in the *late-video* case. In addition, we concluded that the sequence-bias fusion methods examined – i.e. *ibias* and *sbias* – are not as effectively applicable in audiovisual video classification, as in the image description task. In addition, we verified the complementary relationship between the visual and audio modality, with the majority of multimodal approaches outperforming the best single-modality runs.

For a detailed discussion on the conclusions above, see sections 3.6.4.2 and 3.6.4.3. In general, the acquired results can be interpreted as a baseline performance, achievable by the utilization of the audio and visual modalities with each of the proposed classification models presented here. Specialization to more complex models, more persistent, dataset-wise fine-tuning and approaches pertaining to each specific video classification task (e.g. human action recognition, event detection, e.t.c.) can improve these accuracy scores further. Possible avenues towards this are presented in the next section.

3.6.6.3 Future work

There are a number of ways this work can be extended. In the future we would like to utilize additional video modalities – such as directly utilizing temporal content, video text metadata or detected high-level objects in the video (e.g. faces or segmentation information) – and investigate their combination in the proposed workflows. This entails an extension of the dual-modality settings operated in this work (i.e. the sequence-bias approaches, which assume a “main” and “auxiliary” modality), towards incorporating multiple information channels in the video. The “main” and “auxiliary” channels on audiovisual classification could be swapped, with the audio content considered the primary modality. This could be applied in datasets where the audio modality is the dominant one, such as Audioset. Furthermore, the best-performing *late-video LC* fusion could be modified in order to address its disadvantages. Namely, instead of training two separate models, the marginal modality models could be combined into a two-stream model with the streams combined via a learnable weight w . Regarding the clip extraction process, instead of random selection of frame sequences the clip extraction could be implemented in a sequential moving window in the video. In this scenario, temporal inter-dependencies could be exploited on the clip level – in addition to the frame level – with temporal fusion strategies being examined on this level as well. Regarding the frame encoding layer, the performance of model vector representations (like fc8 layer of the Alexnet DCNN) could be explored. As observed in the preliminary experiments in this study, these features could provide a good balance between classification performance and computational cost, since they produce low-dimensional but rich vectors. Furthermore, additional sequence fusion schemes could be investigated (e.g. RNN-based encoder fusion [250]), as well as alternative sequential deep neural models such as GRU [114]. Finally, deeper neural models could be used to encode image data, borrowing from the state of the art and recent advances in image recognition (e.g. state of the art approaches such as in [610, 239]), as well as more sophisticated optimization approaches than mini-batch Stochastic Gradient Descent, such as the Adam [312] or Adadelta [702] optimizers.

3.7 Conclusion and Findings

Summarizing the findings from analyzing existing literature of content-based approaches and the body of conducted research for novel applications, extensions and investigations of representation methods on selected studies, we reiterate and emphasize the observations made in section 1.3. Namely:

- Downstream task performance heavily relies on the construction of rich, informative features.
- Arriving at such configurations can be costly in terms of data and compute requirements. Additionally, each task, domain and setting may work best with a different representation approach – as a result, there is an added expensive step of search-

ing for the best representation method for the case at hand, in order to arrive at rich features.

- On the other hand, methods for introducing rich knowledge (e.g. semantic, categorical, relational information) from existing resources in the representation appear straightforward, for the majority of approaches considered in this chapter (i.e., both vector space-based and graph-based features).

Given the above, we shift our attention on the avenue of knowledge-based augmentation, towards representation enrichment. This will be our focus in the next chapters, where we will argue that building rich representations can be facilitated by utilizing existing sources of high-quality, structured and curated information.

4. ENRICHMENT OF REPRESENTATIONS USING PRIOR KNOWLEDGE: AN OVERVIEW

Up to this point we dealt with content-based representation approaches utilized in the literature, as well as in extensions and/or applications investigated for selected studies. In chapter 2 we organized such methods in three broad categories, under the prism of the classification task across data modalities.

In this chapter, we retain the same task/modality setting, but with a new goal: investigating resources of structured knowledge, along with their utilization for representation enrichment. We begin in Section 4.1 by expanding on the need and motivation for utilizing high-level knowledge for enriching machine learning tasks. Subsequently, in Section 4.2 we cover knowledge resources available in the research community that have been utilized for enriching machine learning tasks. We focus on resources organized in a structured format, so that they can be utilized for automatic data analysis and machine learning workflows with little to no preprocessing. We move on by presenting approaches in the related work for representation enrichment in Section 4.3, that utilize the resources presented. Finally, we discuss and summarize the presented material and selected proposed approaches for enriching classification and automatic summarization tasks with the help of structured knowledge (Section 4.4).

4.1 The need for enrichment

A machine learning system has to overcome multiple difficulties in order to facilitate efficient and robust performance. One of the key challenges is clarifying ambiguity and deducing missing information in the input, the lack of which may hinder its decision-making abilities. Further, semantically rich representations built via knowledge infusion may counteract and sidestep multiple obstacles commonly encountered in a Machine Learning system's lifecycle. In summary, such limitations, considerations and challenges that can arise may include:

- Dependence on critical contextual knowledge, disambiguating factors and other crucial sources of information that may be missing from training data. For example, semantic ambiguity in textual passages, unclear scale / orientation of visual objects, auditory pareidolia, etc.
- Incomplete domain-specific knowledge in the training data, such as named-entity information, audiovisual logos and/or artifacts of special importance and/or contextual meaning.
- Factors inherent in the data generation setting. For example, subjective attributes of human generators such as education and writing style, cultural / biographical elements (e.g. prosody and dialects) can severely alter semantics, shifting cultural zeitgeist expressed in texts and multimedia (e.g. memes, slang, etc.).

- Inherent data ambiguity (e.g. in language, leading to diverse interpretation, optical / auditory illusions)
- Need for the predictions and/or decision making process of the learner to be transparent and explainable by experts in the task (e.g. machine learning researchers), experts of the domain the system is deployed in (e.g. medical professionals in AI-assisted diagnosis, medical imaging, etc.) as well as laymen (e.g. users of commercial products). Additionally, AI safety and alignment concerns [18] emphasize the need for clarity and understanding of inner workings and decision making processes of Machine and Deep Learning models.
- Lack of available data (e.g. in low-resource domains and/or datasets), which is a severe limitation when employing data-driven representation learning methods
- Lack of available computational resources. This can severely handicap training, evaluation, tuning and deployment of Machine Learning pipelines, hindering progress in both the industrial and research sectors.
- Desire and/or commitment to use a limited budget of computing power, which organizations may pursue towards a more “Green AI” vision, given the non-trivial carbon footprint of modern Deep Learning systems [599, 335].
- Limitations of data representation technology, resulting in a large semantic gap between engineered or learned input representations and human concepts [574, 47] that even advanced content-based representation extraction cannot address

This set of limitations is something that a learning system has to consider. One avenue of alleviating such problems is the injection of knowledge to the information that is available to the learner, which is an approach for improving machine learning systems that shows promise [578]. In this study, we focus on enrichment with high-quality, structured and curated knowledge, e.g. from resources such as knowledge repositories, databases, ontologies and lexicons. Early attempts of such research efforts relied on rule-based systems [19]. These systems maintained expert knowledge preconfigured into decision rules relevant to the task, serving as an oracle on which the system relied for decision-making. Additionally, knowledge resources were constructed from data, inferring relationships and conceptual information by human experts [86, 145]. This effort, along with the drive to digitize human knowledge towards easier dissemination, led to a wealth of information being available in machine-readable format. As a result, a large number of resources can be readily exploited towards improving tasks for which a relevant and applicable resource exists [28]. In light of these, we move on to examine such resources, along with avenues of utilizing their contents through representation enrichment, towards improving task performance.

4.2 Knowledge Resources

In this section we present a set of resources used for enriching representations for machine learning tasks. We summarize the resources covered in table 4.1, outlining the type

name	unit type	relation	compilation	endpoint / format	url	language
Wordnet [441]	concept	hierarchical, semantic	manual	multiple / multiple	web , nltk	multiple
SentiWordnet [29]	concept	polarity, hierarchical, semantic	automatic	python, web / text	nltk, code	english
Framenet [178]	semantic frame	frame-semantic	manual	python / -	web	multiple
Babelnet [456]	concept / entity	hierarchical, semantic, relatedness	automatic	multiple / multiple	web	multilingual
DBpedia [353]	property-value	linked-data	mixed	web, SPARQL, REST / RDF, JSON	web, code	multiple
Wikidata [645]	property-value	linked-data	manual	multiple / multiple	web	english
ParaphraseDB [190]	phrase	paraphrasal	automatic	web / text	web	multiple
Freebase [68]	property-value	linked-data	manual	web / text	web	english
YAGO [603]	entity	hierarchical, semantic	manual	REST, web / JSON	web	multiple
ConceptNet [380]	concept	hierarchical, semantic, similarity	mixed	python, REST, web / JSON	web	multiple
Cyc [355]	concept / entity	hierarchical, semantic	manual	multiple / multiple	web, code	english
Imagenet [143]	concept	hierarchical	manual	web / XML	web	english*
AudioSet [194]	concept	hierarchical	manual	web / JSON	web, web	english*
Music Ontology [516]	concept	semantic	manual	multiple / multiple	web	english*
E-ANEW [658]	lexicon	affective	manual	web / csv	web	english
General Inquirer [536]	lexicon	affective	manual	java, python / -	java, python	english
Labelsets	concept	hierarchical, semantic	-	-	-	-

Table 4.1: Knowledge resources for enrichment of machine learning tasks. For library-based programmatic resource endpoints and labelset structure, the format is irrelevant and is omitted. In the language column, entries with a asterisk superscript* refer only to the language in which the resource elements are described in – i.e., the resource content itself is not linguistic and does not lend itself to a specific language.

information units and relational structure, compilation-related information, relevant URLs, as well as the language and format it is available in.

A widely used resource is Wordnet [441], a directed acyclic graph, where each node contains sets of synonymous semantic concepts, or “synsets”. A synset comes with a number of possible strings (i.e. “lexicalizations”) that can be used to express / convey the meaning encapsulated in the synset. Synsets are linked with relations denoting hypernymy / hyponymy (e.g. *dog* “is-a” *animal*), meronymy (e.g. *wheel* “is-part-of” *car*), and others. Additional information is provided, such as synset definitions, example sentences depicting usage of the sense by lexicalizations, POS information, etc. Wordnet was manually compiled, and can be accessed via multiple frontends, including an online interface and a variety of libraries and packages. Sentiwordnet [29] is a lexical resource targeted for aiding sentiment analysis and opinion mining tasks. It is built from Wordnet by automatic annotation of synsets with polarity scores on three axes, i.e. “positivity”, “negativity” and “neutrality”, with each value lying in the range of [0.0, 1.0] and all scores adding to unity. It is available in NLTK [558] and as raw text on the web. Framenet [33] is an English lexical database, annotated with descriptions of semantic roles pertaining to events, relations, situations, entities and related participants, based on frame semantics [177]. It is constructed from the bottom-up, with frames and its contents being organized in a multiple-inheritance structure. For example, the frame *driving* with elements *driver*, *vehicle*, etc. inherits the frame *transportation* with corresponding elements *means*, *path*, etc. Framenet is accessible via the NLTK. Babelnet [456] is a semantic network that incorporates both semantic and encyclopedic information. It builds mappings from disambiguated Wikipedia lemmas and encyclopedic content, as well as sense information from Wordnet. Wikipedia page disambiguation is performed by considering lemma categories, links and sense labels, while for Wordnet, graph node relation linkage and synset definition content are used. Additionally, machine translation techniques and cross-lingual links from Wikipedia

are employed, to provide Babelnet with multilingual information. An API is provided for a number of programming languages.

The CyC project [355] consists of a knowledge base that catalogues a large number of formal assertions that represent existing human knowledge. Facts are encoded into a hand-crafted knowledge representation language (CycL), using a logical framework and a multi-level ontology developed by experts. Additionally, an inference engine is developed in the project. Proprietary and open-source versions of the resource are available, providing multiple endpoints and formats. Further, ConceptNet [380] is a graph database that aims to capture knowledge via the use of semi-structured natural language fragments. In ConceptNet, graph nodes represent concepts both simple (i.e., similar to Wordnet synsets) and compound higher-level entities and events. The latter are built by combining primitive building blocks like verbs, noun phrases and prepositional phrases (e.g. *eat lunch*, *in the evening*, etc.). Nodes are linked via relations expressing synonymy, meronymy, causality, affect, and others. A subset of relations encoded should be viewed as reflecting probabilistic, rather than strict associations (e.g. representing relations like “it often holds that”); in that sense, information in ConceptNet represents informal, everyday knowledge. ConceptNet was built by applying rule-based extraction on crowdsourced “fill-in-the-blank” commonsense sentences. The resource offers a REST API, along with dumps containing the entire graph.

DBpedia [353] is a knowledge base consisting of structured content from Wikipedia. It includes information from article infoboxes, using both manual and automatic rule-based extraction, named-entity recognition and statistical information. It is available in a semantic web structure, using linked data technologies that facilitate cross-lingual cataloguing of Wikipedia literals and property-based relations. The DBpedia project includes an ontology built with the OWL framework. The content of the knowledge base is structured in the RDF format, accessible via SPARQL APIs on the web or obtained via data dumps and the dbpedia-spotlight [128] client. Wikidata [645] is an open collaborative platform, aiming to provide the data available on Wikipedia into a structured and easy-to-use format. The resource provides content ranging from simple property-value pairs (e.g. *author - George Orwell*) to more complex contextual relationships via the use of “qualifiers” (e.g., conveying temporal information) and offers a variety of interfaces and libraries to access its data. The YAGO project [603] combines information from Wordnet and Wikipedia. It consists of an ontology with entities being connected via relations to other entities and literals (such as string lexicalizations and numbers). Relations include real-world associations (e.g. “hasWonAward”) as well as semantic, linguistic and meta-relations (e.g. expressing synonymy, subsumption, type, relation/class hierarchy, etc.). YAGO is built automatically by applying a wide range of heuristics towards information extraction from Wikipedia and subsequent semantic linkage to Wordnet. The resulting data and structure are finalized via a post-processing phase for quality control purposes. A query engine accompanies the resource, where users can extract information, relations, etc. and the entire database as well as domain-specific subsections are publicly available.

Freebase [68] is a collaborative database consisting of a large number of entities, properties and assertions. Entries are organized in tuples in a graph data store. The knowledge

base emphasizes on scalability, collaborative maintenance and ease of use, towards research and open data-oriented community applications. Access to Freebase has been facilitated through Metaweb Query Language (MQL) queries, via an HTTP API with JSON serialization. Currently, Freebase is available via data dumps. ParaphraseDB [190] is a database containing paraphrase pairs, i.e. pairs of semantically equivalent, but syntactically and/or lexically different phrases. The pairs are built by analyzing parallel corpora using the bilingual pivoting approach, where, if different phrases from a source language L_1 translate to the same string in a foreign language L_2 , then they are a paraphrase of each other. The method is subsequently refined via a distributional reranking step that takes into account contextual information of the phrase. The database is available in multilingual versions covering a total of 23 languages [189] and can be obtained from the web in plaintext format.

Multiple resources are organized in a lexicon format, providing information in a mapping organized by name-value correspondence. E-ANEW [658] is a lexicon that provides affective norms for a collection of close to 14000 English words, which is expanding on previous work [78]. The norms compiled in the resource are the following: *valence*, which refers to the degree of pleasantness communicated by the word, e.g. from extremely negative such as “leukemia”, to very positive, like “fantastic”. *Arousal* is correlated with emotional intensity associated by a word (e.g. “dull” to “insane”). Finally, *dominance* is described as pertaining to the degree of power and control exerted by the word (e.g. “dementia” to “completion”). The data in the resource is available in csv format, as supplementary material to the study. General inquirer [536] is a psycholinguistic model that maps affect in spatial coordinates. It contains over 8000 English words, mapped to 182 affective categories that emphasize and focus on coverage of a broad range of psychological states. The lexicon is available in java and python frontends.

There are some knowledge resources that deal with multimedia. The Imagenet [143] dataset links 80000 Wordnet synsets to high-resolution representative images, including animals, objects, scenes, etc. It retains the semantic hierarchy structure of Wordnet and provides a few hundred images per concept, resulting in a semantic graph mapped to approximately 3.5 million images in total. The data was compiled through automatic image retrieval methods, refined by quality control with crowdsourced means. Additionally, object-level annotations (i.e., bounding boxes) for object detection are provided. Commonly used subsets are available as data dumps, hosted and/or mirrored on the project’s website. The Audioset ontology [194] is an audio knowledge base, organizing audio categories into a hierarchical structure, with abstract classes such as “human sounds”, “music”, “natural sounds” and more refined categories such as “whistling”, “musical instrument” and “wind”. Each ontology item is accompanied by a textual name, a short textual description and a URL link to a video that contains audio representative to the category. The resource is available in JSON format, while the dataset version (that includes the video links) are available in CSV. The Music Ontology [516] is built utilizing previous work of the authors in timeline and event-related ontologies, introducing musical concepts ranging from music “Composition” and “Arrangement” to “MusicalWork”, “Recording” and “Signal”. The ontology introduces three levels of granularity and expressiveness, from editorial information

(e.g. track / artist / album information), through performance-related concepts (e.g. with a granularity such as performance / recording / audio stream concepts), to decomposition to fine-grain musical elements (e.g. key, musical instrument, temporal localization of an audio piece). Multiple formats are available on the project’s website.

Additionally, exploitable information can be extracted from the ground truth / labelset of the data. For example, dependencies and relationships present in hierarchical labelsets: in a label hierarchy organized in a tree, a low confidence for a superclass automatically provides a bias against predicting its children. This can be exploited by, e.g., hierarchical classification approaches [222, 575], but also is a form of knowledge that can be integrated in the representation of corresponding data. In addition, ground truth information and metadata accompanying the data labelset can be utilized in the representation construction phase, such as user tags, textual descriptions of multimedia instances, etc. Finally, along with the presented resources, there are tools and approaches that build, expand on and merge knowledge repositories and distributional information [142, 270, 173, 61, 371] to arrive at highly specialized and adaptable information sources.

Having described available knowledge resources that can be utilized for enhancing representations for machine learning tasks, we move on to present related work that have implemented such approaches with a variety of enrichment approaches and for different data modalities.

4.3 Representation enrichment approaches

Here we combine the material covered in previous sections and chapters – i.e. representation approaches from chapter 2, and knowledge resources / representation enrichment motivation from section 4.2 – to a presentation of methods in the literature that inject knowledge into data representations, in order to enhance machine learning tasks for different modalities.

Comprehensive literature reviews for knowledge integration in machine learning tasks include the study in [16] that focuses on quantitative comparison of the experimental results of a broad range of approaches for text classification. Further, the work of [627] investigates semantic variants of the well-known approach of vector space models [545], and the utilization of the resulting vectors in text machine learning tasks. In [88], the authors focus on building sense representations of text in both supervised and unsupervised settings, along with application and evaluation approaches. The survey in [176] covers symbolic, distributed and distributional representations, focused on NLP tasks. The survey in [70] provides a brief overview on external knowledge that is expressed in constraints, focusing on deep learning approaches.

We separate and organize the outlined work in three broad categories, depending on how the enrichment is applied in the representation extraction and/or learning process.

- Input modification, covering approaches that insert external knowledge at the input

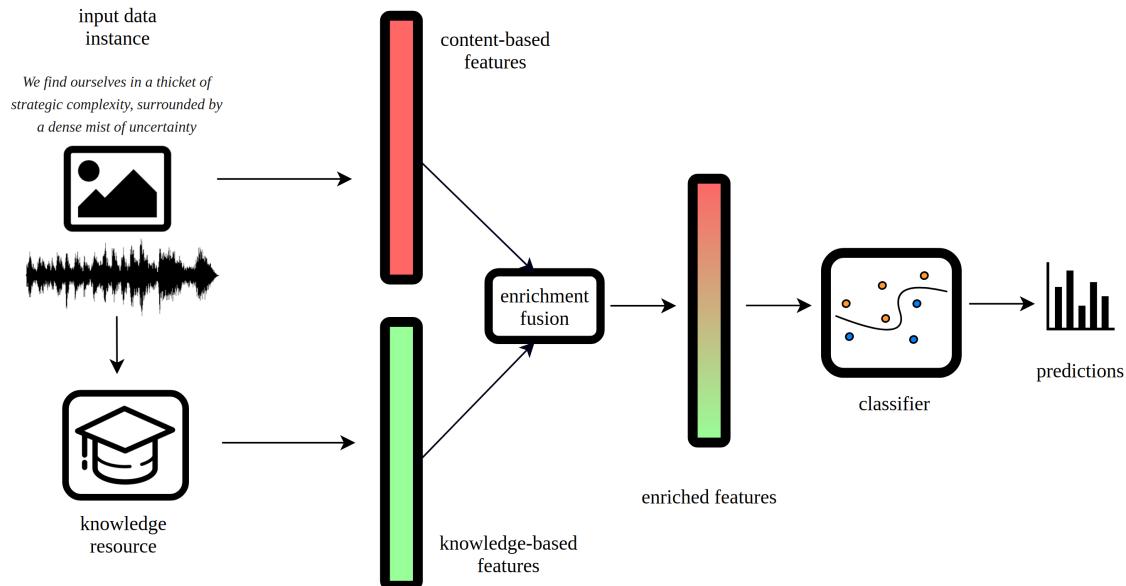


Figure 4.1: Input modification enrichment strategy: Content-based feature information from an input instance (text, image or audio) is combined with knowledge-based features via various fusion methods (e.g. concatenation, averaging, replacement, etc.). Red and green colors represent content-based and knowledge-based information, respectively. The resulting enriched features contain information from both sources, hence the mixed coloring. Knowledge-enriched features are subsequently fed to the classifier.

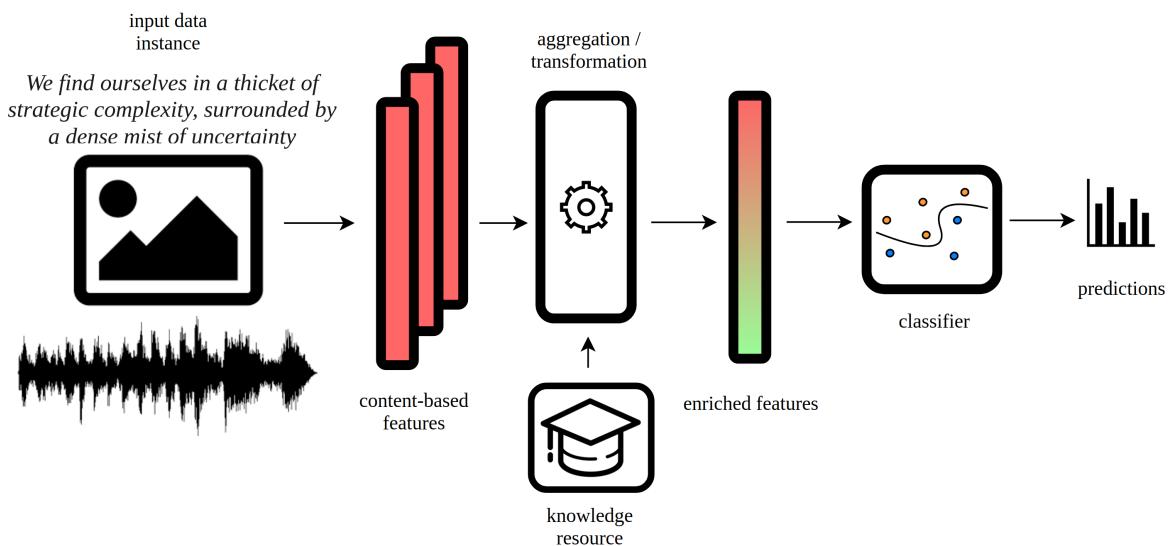


Figure 4.2: Knowledge-based refinement strategy: Aggregation, transformation and/or modification of content-based features from an input instance (text, image or audio) is guided / informed by knowledge-based information. The resulting aggregated and knowledge-enriched features are fed to the classifier.

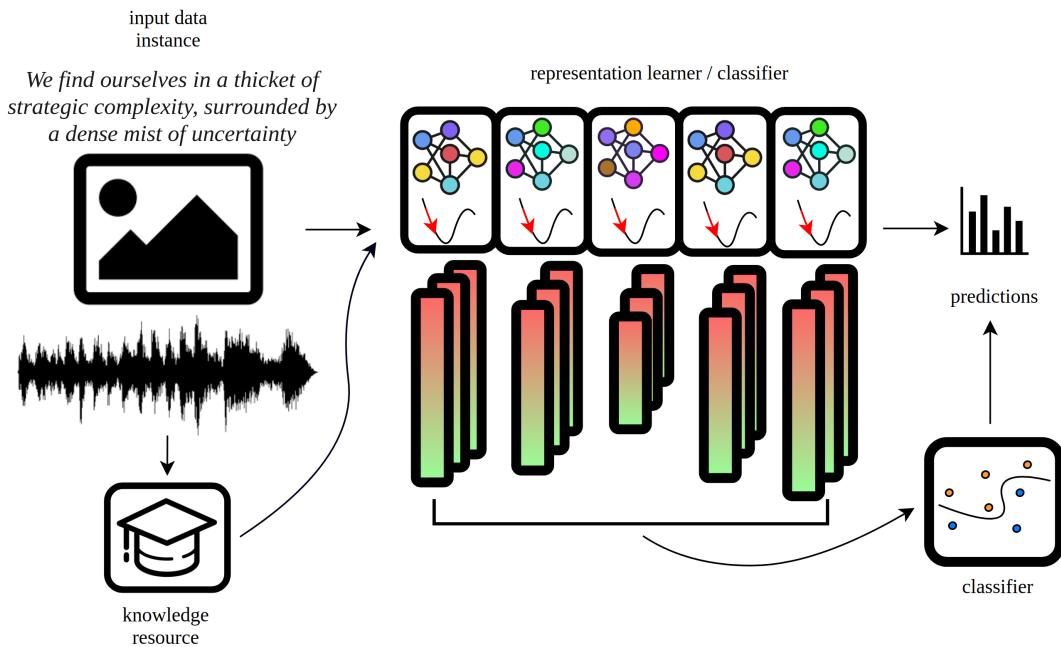


Figure 4.3: Knowledge-aware end-to-end systems: Content-based and knowledge-based information from an input instance (text, image or audio) is jointly considered to build deep knowledge-enriched end-to-end models.

feature level, arriving at a configuration where knowledge-based information is fed as an input of the machine learning pipeline. A visualization is available in figure 4.1.

- Representation refinement, which includes methods that transform / modify / process existing low-level representations in a manner that is directed and guided by external information (rather than content-based). See figure 4.2 for a visualization.
- End-to-end knowledge-aware systems, where deep hierarchical architectures are built from both content-based and knowledge-based information (illustrated at figure 4.3).

The following sections elaborate on each category. We argue that this grouping bears meaningful analogies to the categories we adopted in the presentation of non-enriched representation avenues (i.e. chapter 2). Table 4.2 lists the approaches presented in the paragraphs that follow, conveying similar information as described in Section 2.2. Additionally, the table includes a categorization over the enrichment approach adopted per study, along with the resource that facilitates the supply of external knowledge. In cases where the resource is not widely used, we refer to the type of knowledge the enrichment is facilitated with.

4.3.1 Input enrichment and modification methods

citation	mod.	enrichment - resource	category	representation	labelling	classifiers	metrics
[158]	TXT	input - Wordnet	LOW	TF-IDF	MC-SL/ML	similarity	F1
[462]	TXT	input - Wordnet	LOW	TF-IDF	MC-SL/ML	SVM, DT, KNN	P
[503]	TXT	input - Wordnet	DEEP	CBOW, BoW, TF-IDF	MC-SL/ML	MLP	F1
[582]	TXT	input - Wordnet	LOW	BoW, TF-IDF	MC-SL	SVM, MLP, LSTM	F1
[330]	TXT	input - SentiWordNet	AGGR	PCA, LSA, GI, GR	MC-SL	SVM, NB, k-NN	ACC
[363]	TXT	refinement - Sentiment lexicon, synonyms	LOW	glove	MC-SL	SVM, Adaboost, NB	F1
[693]	TXT	refinement - E-ANEW	DEEP	skipgram, glove	MC-SL/ML	CNN, DAN, LSTM	ACC
[571]	TXT	refinement - Paraphrases	DEEP	ELMO	BIN	MLP	ACC
[215]	TXT	refinement - Wordnet, synonyms/antonyms	DEEP	skipgram, glove, fasttext	MC-SL	similarity	ACC
[607]	TXT	E2E - Entities (Dataset)	DEEP	TRANSFORMER	BIN	NEURAL	ACC
[714]	TXT	E2E - Entities (TAGME, Wikidata)	DEEP	TRANSFORMER	MC-SL	NEURAL	ACC
[497]	TXT	E2E - Wikipedia, YAGO, Wordnet	DEEP	TRANSFORMER	BIN/MC-SL	NEURAL	ACC
[306]	TXT	E2E - Sentiwordnet	DEEP	TRANSFORMER	MC-SL	NEURAL	ACC
[101]	TXT	E2E - Wordnet	DEEP	BILSTM, glove	MC-SL	NEURAL	ACC
[402]	IMG	input - Labelset	LOW	color, texture	MC-SL	BN	ACC
[55]	IMG	input - Wordnet, tags	AGGR	TFIDF, color	MC-SL	NB, SVM, BN	ACC
[644]	IMG	input - Labelset	LOW	color, direction, signal	MC-SL	SVM	ACC
[416]	IMG	input - Wordnet, tags	AGGR	SIFT, CLUST	MC-SL	SVM	ROC
[315]	IMG	input - Wordnet, tags, Wikipedia	LOW	MPEG-7	MC-SL	evol. SVM	P, R
[62]	IMG	input - Labelset	AGGR	SIFT, CLUST	MC-SL	SVM	ACC
[593]	IMG	refinement - Wordnet	AGGR	color, texture, shape, CLUST	MC-SL	NB	P, R
[361]	IMG	refinement - Wordnet	AGGR	color, texture, shape, CLUST	BIN, MC-SL	LM, SVM	P, R
[667]	IMG	refinement - region	AGGR	SIFT, CLUST	MC-SL	SVM	AUC
[362]	IMG	refinement - Metadata	DEEP	Decaf, TF	MC-SL	SVM	mAP
[146]	IMG	refinement - Imagenet	LOW	GIST	MC-SL	similarity, SVM	ROC
[414]	IMG	E2E - Wordnet	DEEP	CONV	MC-SL	NEURAL	mAP
[706]	IMG	E2E - Wordnet, Labelset	DEEP	CNN	MC-SL	NEURAL	ACC
[259]	AU	input - Wordnet, E-ANEW, GI	LOW	TFIDF, signal, spectral, MFCC	MC-SL	SVM	ACC
[710]	AU	input - Labelset	LOW	signal, spectral, psych.	MC-SL	HMM, rules	ACC
[90]	AU	input - Labelset	LOW	MFCC, spectral, psych.	MC-SL	k-NN	ACC
[278]	AU	input - E-ANEW, Wordnet	LOW	psych., musical	MC-SL	k-NN	ACC
[477]	AU	refinement - signal proc.	LOW	signal, spectral	MC-SL	kSVM	ACC
[57]	AU	E2E - Wordnet, SentiWordnet	LOW, DEEP	MFCC, spectral, signal, psych., word2vec	MC-SL	k-NN	ACC
[287]	AU	E2E - Labelset	DEEP	CONV, SIAMESE	MC-SL/ML	NEURAL	ACC
[606]	AU	E2E - Labelset	DEEP	CONV, LSTM	MC-SL/ML	NEURAL	F1

Table 4.2: A cataloguing of representation enrichment methods for machine learning tasks, using external information sources. MC, SL and ML labelling refers to multiclass, single-label and multi-label configurations respectively. The enrichment-resource column contains entries in the form ENR-RES, where ENR refers to enrichment approaches described in section 4.3 and RES to the resource / information source or information type utilized for knowledge injection. Entries in the representation / classifier columns refer to acronyms described in the text, or descriptive categories of families of approaches. Evaluation measures ACC, PR, RE, F1, AUC and AP refer to accuracy, precision, recall, F1-measure, area under curve and average precision, respectively. Classifier NEURAL refers to applying the neural model itself with an appropriate output layer(i.e. a d -dimensional fully-connected linear layer followed by a softmax function, for the prediction of d output classes).

4.3.1.1 Overview

A straight-forward avenue towards knowledge-rich learning approaches is injecting knowledge into the feature set extracted from the input data. This entails modifying the collection of features fed to the learning model with high-level semantic, statistical and conceptual information that is relevant to the input. Such information can be extracted by utilizing structured knowledge resources in order to match the input instance to relevant pieces of high-level information.

4.3.1.2 Approaches

For example, regarding classification over textual data, the authors in [158] combine bag-of-words features [544] with concept-based statistics from Wordnet [441], weighted via TF-IDF. They explore different ways for mapping lexical terms to Wordnet concepts (e.g. including expanding to hypernyms) and fusing the lexical and semantic channels into a single vector space representation. Additionally, a feature selection phase with the chi-squared statistic [125] is employed. The final feature vector is used to classify news and online discussion articles via cosine similarity scores. A similar approach in [462] improves upon concept lookup in Wordnet, by utilizing multi-word querying as well as using POS information from Treetagger [555]. Additionally, the authors expand relevant concept mappings by considering multiple semantic relations (e.g. hypernymy, antonymy, troponymy, etc.), as well as investigating different weighting schemes for each concept in the resulting collection. The final vectors utilized are composed of both lexical and semantic information in TF-IDF vectors and are used to categorize news and online forum articles via SVM, Decision Trees and k-NN models. The work in [503], combines concept statistics mined from Wordnet with CBOW embeddings [440], evaluating different methods for lexical-semantic fusion, disambiguation and concept weighting. Additionally, concept mapping expansion is implemented with spreading activation [117], diffusing semantic weights in a controlled manner with each expansion step. The enriched vectors are fed to a feed-forward NN to classify data in various domains (news articles, forum posts, biomedical documents, etc.). The work in [330] uses SentiWordNet to extract four sentiment feature statistics from song lyrics for mood classification. They are combined with multiple transformation and feature selection methods and fed to three different classifiers (NB, SVM, k-NN). In [582], document-level taxonomies are extracted from Wordnet by utilizing hypernymy relations after disambiguation. The final semantic vectors are built via double-normalized TF-IDF vectors [412] followed by the application of statistical, graph-theoretic and ranking-based feature selection approaches, arriving at a fixed representation dimensionality. The final semantic vectors are concatenated with lexical information derived from various BoF and TF-IDF vectors, with respect to different elements of interest in the text. The combined representation is fed to linear SVMs, MLPs and hierarchical attention networks [686] in order to predict gender, personality type, age, news topics, drug side effects and drug effectiveness, utilizing different domains and datasets.

For images, many approaches take advantage of annotation hierarchies and label seman-

tics, as well as image-level localized annotations. In [402], ground truth annotations as well as dedicated classifier predictions of regions marked as “sky” or “grass” in an image are employed for indoor / outdoor classification. This information is combined with low-level color and texture features to produce the final result via a Bayesian Network (BN). In [55], aggregated color features along with textual information (mapped to TF and TF-IDF vectors) are utilized from images and tags, subsequently clustered into a multimodal knowledge graph of mid-level concepts. Wordnet is used to disambiguate word senses and facilitate semantic similarity extraction between concepts in the built graph. Classification is performed using a two-staged prediction process: a concept prediction phase with SVM and NB learners, followed by the application of a Bayesian Network (BN) over the concept-level predictions. The approach is used to generate predictions on plants, animals, people and landscape labels. The authors in [416] take advantage of encapsulated relations in Wordnet, such as hypernymy, meronymy and holonymy to extract training examples for training hierarchical classifiers. Examples are retrieved for training via textual tags / annotations in the image, while pruning methods are applied to inhibit propagation towards too generic graph nodes. The extracted relationship subgraph is used to train arrays of binary SVMs for predicting different labels pertaining to animals, people and vehicles, using a visual vocabulary clustered from SIFT features on Harris-detected keypoints. The work of [315] enhances image segment-based predictions with keyword-based, conceptual information. Image tags are converted to entities via a named entity recognition procedure: each entity is mapped to Wordnet concepts that are most similar to the target image labelset by a targeted hypernym discovery process. Special measures are introduced to improve coverage for hypernym extraction (i.e. tags are expanded by retrieved most relevant Wikipedia articles in terms of textual similarity and article linkage). Regarding visual features, images are segmented with self-organizing maps [317] and particle swarm optimization [307] to regions, out of which low-level MPEG-7 features are extracted and classified to a variety of objects and scenes by an SVM combined with evolutionary methods [483]. An array of pre-selected concepts is used in [644], with training images split into annotated regions of a fixed size. Concept classifiers are subsequently trained to detect these concepts, with the generated conceptual model vectors being utilized for the final step of scene classification. SVM classifiers are used for both detection levels, with the conceptual vectors being concatenated with multiple low-level features such as color, direction and intensity histograms. The work in [62] utilizes the VOC2006 taxonomy [170] as the prediction target. The actual / predicted semantic class similarity is used in the training loss, quantified by the number of nodes between the actual and predicted class in the taxonomy. The ensemble consists of multiple one-vs-rest binary SVM classifiers, jointly trained to predict the 10 categories in the taxonomy. SIFT features are used for visual content representation, clustered into visual codebook with k-means.

Regarding audio data, ground truth information as well as textual metadata have been used for enrichment in the input level. The work in [259] exploits audio and textual features for music mood classification. Wordnet, Wordnet-Affect, General Inquirer and E-ANEW are utilized to extract semantic and psycholinguistic information from song lyrics. These are combined with statistical, POS and heuristic-based features from the lyrics, mined as bags of word n-grams in boolean or TF-IDF weights. Various configurations of feature

combinations are investigated. A variety of MFCC, spectral and statistical features are utilized for audio processing, while SVMs with linear kernels were the classifier of choice. The work in [710], a coarse-grained classification is used as an intermediate high-level representation. Audio-oriented classes such as silence, speech, music, as well as multiple types of environmental sounds are used, prior to categorization into finer classes (e.g. as rain, bird sounds, etc.). Signal-based, perceptual features and clustering are employed, while HMM learners are utilized for classification. In [652], spectral and musical audio features . In [278], E-ANEW is used to extract valence and arousal estimates from song lyrics, towards emotion classification. The resource E-ANEW is expanded via synonym extraction through Wordnet, with POS information being used to tackle ambiguity. Musical and psychoacoustic features are used for audio description and classification is facilitated via k-NN matching. Finally, the work in [90] utilizes Wordnet at the annotation level. Audio samples with Wordnet annotations (i.e., links to audio-related Wordnet synsets) are utilized for training a classification system to predict concept-level confidence scores. MFCC, statistical and spectral features are utilized for audio representation, fed to a k-NN classifier in order to classify instrument sounds.

4.3.2 Knowledge-based refinement methods

4.3.2.1 Overview

Another avenue of introducing external information is by modifying the construction procedure of building and/or learning input data representations. Here, bias generated from structured knowledge can be useful in the representation and can be introduced in a variety of ways; for example, aggregation and grouping criteria of early / low-level features can enforce relationships and associations of input data, encoded in existing knowledge resources, as filters and membership indicators to inform grouping and partitioning. Additionally, representation learning methods can utilize semantic relations to inject constraints, regularization terms and scaling factors into the loss function, similarity function or fitness objective. For example, taxonomic information linking together the concepts “dog” and “wolf” (i.e. as subspecies of “Canis”) may be exploited into embedding generation methods, e.g. by encouraging the respective data points to lie close in the embedding space. Further, a database of synonyms can be utilized by, e.g., a bag-of-words approach, in order to merge or apply special weights to data instances of the concepts “dog” and its synonym “Canis Familiaris”. These modifications are informed and quantified by interacting with the external resource and utilizing relationships encompassed within it, leading to the production of knowledge-enriched features.

4.3.2.2 Approaches

Regarding refinement approaches in text classification, the approach in [363] expands the distributional model of [493] to consider sentiment, using prior information on two levels: word-level sentiment information (mapping words to *positive*, *negative*, *neutral*, using

multiple lexicons and synonym expansion to achieve high vocabulary sentiment coverage) and document-level information (sentiment annotations on entire documents). Using this information, embeddings are trained and used for sentiment classification with SVMs, Adaboost [183] and NB classifiers. The work in [693] introduces refinement of word vectors in a two-stage reranking scheme: first, the 10 nearest neighbours of a word is retrieved, followed by a reranking with respect to the degree of positive and negative sentiment expressed by each. The cosine similarity between word vectors is used to obtain semantic similarity, while sentiment score is provided by the “valence” norm in the E-ANEW [658] lexicon. The refined output of GloVe vectors [493] and SkipGram embeddings [440] are evaluated in binary and multiclass sentiment classification via feed-forward (CNN [311], DAN [273]) and recurrent (bi-LSTM and TreeLSTM models [612, 393]) neural classifiers. In [571], the contextualized ELMO embeddings[496] are retrofitted with paraphrasal information. The approach learns an orthogonal transformation into a vector space where projections of a sentence and its (semantically similar) paraphrase are collocated. The resulting enriched sentence embeddings are evaluated in multiple binary sentiment analysis tasks, via an MLP classifier. The work in [215], seek to maximize and minimize antonymy and synonymy constraints respectively, while trying to maintain distances between remaining instance pairs. The authors investigate different optimization objectives, knowledge resources (i.e., Wordnet and Roget’s Thesaurus [279]) and source embeddings (SkipGram, GloVe and FastText) to train a fully-connected neural network as the mapping function. The model is evaluated on multiple settings, including classification tasks such as word replacement for lexical text simplification, where replacements are retrieved via maximal similarity.

Regarding image data, research efforts have to deal with a larger semantic gap compared to textual data; for the latter, high-level semantic concepts can be retrieved from words, sentences or paragraphs: these are data tokens that result from semantic segmentation performed by humans, readily available and delineated in the grammatically sound text. In contrast, content-based image approaches can only extract primitive features from pixel intensities with little to no semantic content or direct linkage to high-level information [569].

The work in [593], builds a semantic hierarchy of Wordnet senses from image keywords, resulting in an “ontology-induced visual vocabulary”. This is achieved via a region-to-concept classification, mapping image patches concepts which are expanded by Wordnet semantic relations. Weighted K-means is then employed to quantize the regions into a vocabulary, with images being represented via low-level features such as color, position, texture, shape, etc. A hierarchical NB variant [421] is used to classify images depicting a variety of concepts and scenes. A similar approach is applied on the same visual domain in [361], where a visual vocabulary is built via K-means from image regions, each associated with a keyword. Constraints and semantic relations from Wordnet inform the clustering objective, which is modified to consider a semantic similarity component between keywords. A hierarchical ensemble model is utilized with an array of binary Gaussian SVMs to predict image keyword labels and a language model operating on statistics of predicted keywords. In [362], the authors adopt a multi-instance learning approach with privileged information [637], using retrieval methods to obtain training samples from the

web. For training data, metadata in the form of textual descriptions of retrieved images are exploited, being encoded to term-frequency feature vectors, while convolutional Decaf features are used for visual content description [153]. The textual component features are used to modify the objective of an SVM, and the classifier is evaluated in intra and inter-domain settings. The work in [146] uses a visual similarity measure based on the GIST descriptor [472], along with semantic similarities derived from Imagenet. The latter are computed by extracting the k visually nearest neighbours in the Imagenet graph of each image in an input pair. Subsequently, a finalized semantic similarity score is measured from pairwise neighbour similarities or the overlap of their category distributions. The purely visual and semantic similarity scores are then combined in a distance learning scheme with a linear SVM to classify Imagenet object classes. In [667], object and foreground detection ground truth information is exploited, with features extracted from an object region being grouped together as semantically similar. These features are encouraged to be mapped to the same visual words via a distance metric learning process in the proposed semantics-preserving BoW model. SIFT clustered into codebooks of visual words is used for representation, using different distance metric learning techniques such as neighbourhood component analysis (NCA) [217]. The prediction scores are produced by SVMs to classify urban environment objects.

Regarding refinement approaches in the audio domain, our investigation discovered very few suitable studies that do not match the other enrichment categories proposed. In [477], knowledge about audio signal processing is encoded into “Analytical Features”, a framework encapsulating knowledge over acoustic feature design (e.g. heuristics, patterns, etc.). Audio feature engineering paradigms are formed into operators that process / transform / compose different signal / spectral-based audio information. The framework is evaluated over fine-grained categorization for artificial and natural sounds (e.g. dog bark classification, percussion classification) with a polynomial SVM.

4.3.3 Knowledge-aware end-to-end systems

4.3.3.1 Overview

Adhering to the popularity and performance of large end-to-end learning models, many approaches utilize information resources in conjunction with deep learning and neural networks. Here, the injection of external information may occur at the input and/or refinement stages of the pipeline (as in the previous enrichment categories examined). The distinguishing factor is that content-based and external information is jointly exploited in an end-to-end fashion, to automatically learn knowledge-enriched feature hierarchies. Additionally, such approaches often jointly learn the representation and the discrimination/prediction component that performs the downstream task, and/or arrive at representations that are robust across domains and datasets via transfer learning.

4.3.3.2 Approaches

In the text domain for classification, the knowBERT approach [497] offers a method to integrate pre-trained language models with knowledge bases structured in triplets or graphs, given that entity extraction and entity embedding generation mechanisms are defined. Self-attentive mention span embeddings are computed for each candidate entity, followed by neural entity linking [318, 352]. The lexical embeddings are re-contextualized with the produced knowledge-rich entity-span vectors, using a multihead attention transformer layer. Wikipedia, YAGO and Wordnet are examined for entity identification, with embeddings built via the SkipGram algorithm. The model is applied to classification tasks such as binary word sense and subject-object relation classification. The ERNIE model [607] expands on the BERT architecture [148] by jointly utilizing information from knowledge bases alongside distributional information from text. This is introduced by knowledge-level masking: specifically, phrase-level and conceptual/named-entity-level masking is adopted, extending the byte-pair encoding (BPE) masking approach of BERT. This modelling approach is utilized in pretraining, which is performed on multiple-domains and heterogeneous data. The constructed embeddings are used to predict binary sentiment scores on Chinese texts discussing to hotels, books and electronic computers. Another ERNIE model [714] focuses on named entity information, with distinct textual and knowledge encoders handling lexical/syntactic and fine-grained entity-related information, respectively. The TransE model [69] and Wikidata is used to encode entities into vector embeddings, which are combined with lexical token embeddings via multi-headed attention. TAGME [175] is used for entity extraction. Additionally to the pretraining tasks of BERT, the model is also fitted towards entity prediction (similar to training a denoising autoencoder [642]), utilizing special tokens for entities and masked language modelling. The experimental evaluation utilizes the model for word relation classification. SentiLARE [306] uses POS information to extract word-level sentiment polarity from SentiWordnet [29] to train a multilayer transformer network. Masked language modelling is used for pretraining, with context-aware sentiment attention weighing polarities from individual words to the sentiment score of the entire sentence. Pretraining subtasks include predicting both sentence and word-level label information. The model is fine-tuned and evaluated on multi-level sentiment analysis on customer reviews, using Roberta [385] as the transformer architecture. In [101], multiple lexical relations (e.g. synonymy, antonymy, hypernymy, hyponymy, co-hyponymy) are mined from Wordnet in order to enhance premise / hypothesis classification on sentence pairs. The relations are represented as a binary vector, as well as being mapped to embeddings via graph embedding methods (i.e., TransE). The model is composed of two biLSTM layers, for sequence encoding and decoding word relationships, respectively. Semantic lexical information is weighted by the alignment score between word pairs. The system is jointly trained with an MLP classifier layer, and is evaluated on inference sentence pair categorization.

There have been fewer applications of knowledge-aware end-to-end systems for image and audio data, compared to such approaches in the text domain, as well as other enrichment avenues for images and audio investigated in this study. The work in [414] proposes using the Graph Search Neural Network [366] for image classification. The latter operates

on a knowledge graph of candidate image labels, assigning a confidence score to each node, as well as providing a global result. An object detector / classifier is used to identify an initial visual node in the image, from which controlled propagation supplies additional neighbour nodes of visual objects. Wordnet is used to build the knowledge graph, in conjunction with an object detection dataset. The system is applied for multilabel classification over a variety of visual object classes. In [706], an image knowledge graph (IKG) is constructed, containing semantic associations between content in the image to objects and scenes relevant to it. The labels for the latter are produced either by classifiers trained on Imagenet, or semantic associations of the image lexical label, which are extracted from Wordnet. A similarity score based on co-occurrence statistics of objects / scenes detected between image pairs is used to augment learning, utilized in different DCNN models to predict object and scene labels.

For utilization of end-to-end knowledge systems in audio, in [57], audio data and text transcriptions are used in an audio-based / multimodal approach for humour prediction. MFCC, spectral, signal-based and psychoacoustic features are employed for audio representation. For text, content is mapped to word2vec embeddings, bags of ngrams, as well as information related to syntax, sentiment, antonyms and speaker turn. Wordnet and SentiWordnet are used for extraction of semantics and sentiment polarity. CNN, RNN and CRF [336] classifiers and pipelines are evaluated with different combinations of input features. The work in [287] proposes two architectures for ontology-aware sound event classification on audio spectrograms: first, a feed-forward neural network directly models the ontology classes in a sequential manner, i.e. reserving fully-connected layers to produce predictions for each level in the ontology. Each prediction is subsequently fed to the next level as input features. The second approach uses a siamese network [106] to train sample embeddings, where samples that belong to the same class are encouraged to map to vectors that lie closer to each other in the embedding space. The network uses the Euclidean distance, and is trained on triplet instances of all potential cases of a two-layer ontology (i.e. matching subclass, only matching superclass, different superclass), with the final classification being the same as the first architecture. In [606] the authors propose to consider labelset relations via an ontology-aware neural network. The latter consists of a base network of a CNN followed by an LSTM, with feed-forward and graph convolutional networks (GCN) [709] modelling intra and inter-dependencies between levels of hierarchy in the ontology, respectively. Their system is evaluated in single and multi-label classification of urban sounds.

4.4 Conclusion

In this section, we expanded on the motivation for exploiting knowledge resources towards aiding machine learning tasks. Focusing on the classification task for different input data (text, images and audio), we present issues and challenges a classification pipeline has to consider that could be alleviated / informed by existing knowledge. Specifically:

- We covered available and exploitable knowledge resources, along with details re-

garding their information content, structure, availability and retrieval details.

- We presented representation enrichment approaches in the literature, organized in three meaningful categories, with respect to how this augmentation process operates in the feature extraction / learning pipeline: Input enrichment and modification covers methods that inject knowledge-based information in the feature space, providing it as an input to the classification pipeline. Representation refinement methods seek to aggregate and/or transform representations, with these operations being determined, guided and controlled from information that reside in knowledge resources (rather than content-based information or engineered heuristics). Finally, knowledge-aware end-to-end systems adopt deep learning methods to build representation learning and classification machines that jointly learn and operate on data and knowledge inputs.
- We catalogued all covered methods in the literature, including details on enrichment approach, representation, learning, evaluation and label setting

Considering this body of related work, we can observe that there is a lack of investigation of *input-modification* strategies in conjunction with *deep content-based representations*; while input modification has been applied to low-level and aggregation-based representations, we did not discover related work examining it in conjunction with deep content-based features. This is an interesting approach worthy of investigation; it combines the expressive power of deep content-based features, with the simplicity, easy applicability and improved explainability potential of input modification techniques.

Given these remarks, in the next chapter we propose and implement two novel enrichment methods under the aforementioned paradigm, focused on the classification and automatic summarization task for text data. Additionally, we extend our implementation by applying aggregation-based approaches to the knowledge-based features generated for this combination.

5. NOVEL APPLICATIONS AND STUDIES USING ENRICHED REPRESENTATIONS

We now delve into the proposed approach for introducing external semantic information into a learning system, specifically a deep neural network. The proposed method is applied for the classification (section 5.1) and automatic summarization (section 5.2) tasks over the textual modality, and lies in the input enrichment and modification category (as described in section 4.3.1).

5.1 Enriching Embeddings for Text Classification

We begin with enriching text classification, presenting the positioning / motivation of the work in section 5.1.1. We subsequently present the textual (raw text) component of our learning pipeline in Section 5.1.3, the semantic component in Section 5.1.4 and the training process that builds the classification model in Section 5.1.5.

5.1.1 Introduction and Overview

The rise of deep learning has been accompanied by a paradigm shift in machine learning and intelligent systems. In Natural Language Processing applications, this has been expressed via the success of distributed representations [248] for text data on machine learning tasks. These typically come in the form of *text embeddings*, which are vector space representations able to capture features beyond simple statistical properties. Such approaches try to evolve over the histogram-based accumulation used in methods like the bag-of-words model [544]. Instead of applying a hand-crafted rule, text embeddings learn a transformation of the elements in the input. This approach avoids the common problem of extreme feature sparsity and mitigates the curse of dimensionality that usually plagues shallow representations.

There have been numerous approaches to learning text embeddings. Early attempts produce shallow vector space features to represent text elements, such as words and documents, via histogram-based methods [304], [544], [290]. Other approaches use topic modelling techniques, such as Latent Semantic Indexing [139] and Latent Dirichlet Allocation [244]. In these cases, latent topics are inferred to form a new, efficient representation space for text. Regarding neural approaches, a neural language model applied on word sequences is used in [51] to jointly learn word embeddings and the probability function of the input word collection. Later approaches utilize convolutional deep networks, such as the unified multi-task network in [118], or introduce recurrent neural networks (RNNs), as in [439]. Deep neural models are used to learn semantically-aware embeddings between words. [438, 439] These embeddings try to maintain semantic relatedness between concepts, but also support meaningful algebraic operators between them. The popular

word2vec embeddings [437] learn such embedding spaces via Continuous Bag-Of-Words (CBOW) or skip-gram patterns, sometimes varying the context sampling approach [440].

Despite numerous successful applications of text embeddings, most approaches largely ignore the rich semantic information that is often associated with the input data. Typically, such information – e.g. in the form of knowledge bases and semantic graphs – is readily available from human experts. This fact can function both as an advantage and a disadvantage in learning tasks. On the one hand, if the learned model is not restricted by human experts' rules and biases, it is free to discover potentially different (and often superior) intermediate representations of the input data [52] for a given task. On the other hand, ignoring the wealth of existing information means that any useful attribute is captured by relearning from scratch, a process that requires large amounts of training resources.

We claim that we need to investigate hybrid methods, combining the best of both worlds. As such, these methods will allow a model to search the feature space for optimal representations, while being able to exploit pre-existing expert features. We expect such a paradigm shift to affect a multitude of Natural Language Processing tasks, from classification to clustering and summarization. Furthermore, it allows researchers to utilize a full range of different structures over resources: raw text, documents accompanied by meta-data and multimodal components (e.g., embedded images, audio, etc.). However, we expect that the way of introducing semantic information to the model will affect training and the performance of the learned model on the task at hand.

In this context, this work suggests a method to integrate semantic information into a neural classification pipeline, and evaluates the outcome. We model the pipeline as a hybrid, branched architecture (cf. Figure 5.1), where each branch corresponds to one aspect of the hybrid: semantically-driven vs. raw-data-driven.

Given an input text, the semantically-driven branch, elaborated in Section 5.1.4, does the following:

- For each word it extracts semantic information from an appropriate resource of existing knowledge, such as a semantic graph.
- It generates a *semantic vector* for the word.
- It represents the whole text as a fusion of the word semantic vectors.

The raw-data-driven branch, utilizes raw data information to generate a word embedding as we find in many deep learning related works. Finally, we augment the word embedding output representations by the semantic vector, feeding the resulting enriched, hybrid representation to a deep neural network (DNN) classifier.

Based on this formulation, we address the following research questions in the context of the single-label text classification task:

1. Can semantic information increase the task performance, when applied in this setting? If so, how much?

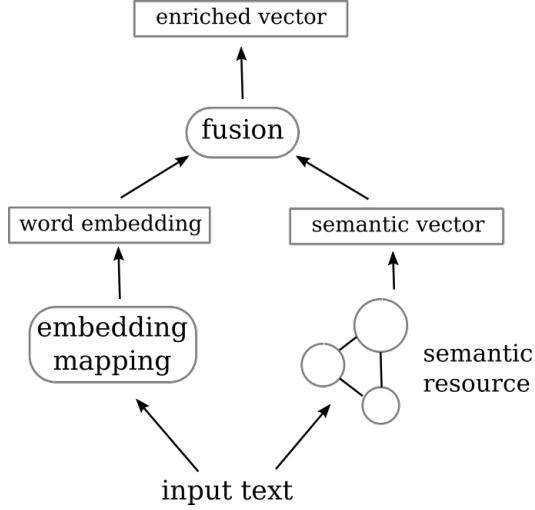


Figure 5.1: Overview of our approach to semantically augmenting the classifier input vector.

2. How much do the different semantic disambiguation methods affect the above performance increase?
3. What is the effect of taking into account hypernymy relations in the semantic branch of the representation?

The rest of the section is structured as follows. Section 5.1.2 surveys relevant works on semantic augmentation methods for classification problems, as well as the main techniques for enriching word embeddings with semantic information. In Sections 5.1.3 and 5.1.4, we elaborate on our method, describing (a) the embeddings generation; (b) the semantic information extraction, and (c) the vector augmentation steps. Section 5.1.7 presents the experimental study that evaluates the performance of our workflow and compares it to the state-of-the-art in the field. It also addresses the above three research questions based on the results. We conclude the investigation with a summary of our key findings along with directions for future work in Section 5.1.9.

5.1.2 Related Work

This study explores the introduction of semantic information into text-based embeddings, in the context of *single-label text classification*. Given a collection of text documents $T = \{t_1, \dots, t_N\}$, a set of c predefined labels $L = \{l_1, \dots, l_c\}$ and existing document-to-label annotations $G = \{(t_1, g_1), \dots, (t_N, g_N)\}$, $g_i \in L$ such that the label of document t_i is g_i , the task is to find a classification function f that produces the correct label for each input document, *i.e.*, $f(t_i) = g_i$.

Specifically, we focus on DNN classifiers in conjunction with word embeddings for representing and feeding the input text to the predictive model.

In the literature, numerous studies leverage semantic knowledge to augment text mining tasks. For classification, graph-based semantic resources such as the WordNet ontology [441] have been widely used to enrich textual information. Early approaches examined the effect of WordNet’s semantic information on binary text classification, using rule-based discrimination [562] as well as SVM classification [413]. In [158], the bag-of-words vector representation [544] is combined with the WordNet semantic graph. A variety of semantic selection and combination strategies are explored, along with a supervised feature selection phase that is based on the chi-squared statistic. The experimental evaluation on the 20-Newsgroups and Reuters datasets shows that the semantic augmentation aids classification, especially when considering the most frequent related concept of a word. Frequency-based approaches are examined in [462] over the same two datasets, applying multiple classifiers to terms, WordNet concepts and their combination. The combined approach yields the best results for both datasets, however (a) it uses handcrafted features for the representation of textual information; (b) it employs shallow methods for classification and (c) it considers subsets of the two datasets.

On another line of research, neural methods are coupled with relationships from the WordNet semantic graph [449], producing a language model where words are represented in a binary tree via hierarchical clustering – rather than deriving such a hierarchy from training data. The resulting model trains faster and performs better than the bag-of-words baselines, but worse than the neural language model of [51].

A bulk of later works modify the deep neural embedding training, with many of them investigating ways of introducing both *distributional* and *relational* information into word embeddings. Distributional information pertains to statistics from the context of a word, while relational information utilizes semantic relationships such as synonymy and hypernymy.

In more detail, the “*retrofitting*” method is used in [174] to shift word embeddings towards points in the embedding space that are more suitable to represent semantic relationships within an undirected graph. This is accomplished by post-processing the existing word vectors to balance their distance between their original fitted values and their semantic neighbours. The experimental analysis demonstrates the resulting improvements on the embeddings in a multilingual setting, with respect to a variety of semantic-content tasks. The retrofitting system is specialized in [215] via a feed-forward DNN that explicitly maps semantic / relational constraints into modified training instances to produce specialized embeddings. The authors report significant gains in a series of tasks that range from word similarity to lexical simplification and dialog state tracking. The study in [694] extends the neural language model of [51] and [438] with semantic priors from WordNet and Paraphrase [190]. Their “Relation Constrained Model” (RCM) modifies the Continuous Bag-of-words (CBOW) [440] algorithm, by modifying the objective function to consider only word pairs joined by a relation defined in the semantic ground truth. Additionally, they explore a joint model where the objective function considers a weighted linear combination of both corpus co-occurrence statistics and relatedness based on the knowledge resource. An experimental evaluation on language modeling, semantic similarity and human judgment prediction over a subset of the Gigaword corpus [487] demonstrates that using the joint model for pre-training RCM results in the largest performance increase,

with respect to standard word2vec embeddings.

In [184], the authors model semantic relatedness by computing the length of the shortest path between WordNet synsets. This is mapped to a word pair by considering the maximum possible distance between candidate synset pairs associated with these words. A scaled version of this length regularizes the cosine similarity of the word pair corresponding word embeddings. Both distributional and semantic information are jointly trained via the ADMM multi-objective optimization approach [76]. The authors evaluate their graph distance measure – along with other WordNet distance approaches – on multiple tasks, such as knowledge base completion, relational similarity and dependency parsing. The overall results indicate that utilizing semantic resources provide a performance advantage, compared to using text-only methods.

In [646], embeddings are fine-tuned to respect the WordNet hypernymy hierarchy and a novel asymmetric similarity measure is proposed for comparing such representations. This results in state-of-the-art performance on multiple lexical entailment tasks.

Furthermore, the authors in [672] use skip-grams with relational, categorical as well as joint semantic biases in the objective function. To achieve this, they model a relation r between entities a and b as vector translation (e.g. $a + r = b$), i.e. modelling both entities and relations in the same vector space. Categorical knowledge is limited to fine-grained similarity scores, after discarding too generic entity relationships. All variants are evaluated on analogical reasoning, word similarity and topic prediction tasks, with the experimental results demonstrating that the joint model outperforms its single-channel counterparts (the semantically-aware networks and baseline skip-gram embeddings). In [59], the authors use external resources to introduce syntactic, morphological and semantic information into the generation of embeddings. Experimental results on analogical reasoning and word similarity sentence completion show that the semantic augmentation is the most reliable augmentation approach, compared to a CBOW baseline, with the other resources producing inconsistent effects on performance. The approach in [403] exploits morphological characteristics by training a recursive neural network at the level of a morpheme, rather than a word, which allows for generating embeddings for unseen words on-the-fly. The authors report large gains on the word similarity task across several datasets. Other approaches to fine-tuning embeddings seek to produce robust feature vectors with respect to language characteristics [533], instead of enforcing explicit semantic relationships.

Some approaches apply their findings on text and/or document classification. The authors in [93] propose a neural topic modelling framework that is able to incorporate metadata such as annotations / tags as well as document “covariates” (e.g., year of publication), with tunable performance trade-offs. Experiments over a US immigration dataset show that this approach outperforms supervised LDA [420] on document classification. In [364], the authors use a document-level embedding that is based on word2vec and concepts mined from knowledge bases. They evaluate their method on dataset splits from 20-Newsgroups and Reuters-21578, but this evaluation uses limited versions of the original datasets.

To tackle word polysemy and under/mis-representations of semantic relationships in the training text, many approaches build embeddings for semantic concepts (or “senses”),

instead of words. Such “sense embedding” vectors are studied in [102], where the authors emphasize the weaknesses of distributional, cluster-based models like the ones in [262]. Instead, they use skip-gram initialized word embeddings, aggregated to sense-level vectors by combining synset definition word vectors from WordNet. Word sense disambiguation (WSD) is performed via a context vector, with strategies based on word order or candidate sense set size, for each ambiguous word. Learning uses the skip-gram objective imbued with sense prediction. Evaluations on domain-specific data (*i.e.* small portion of the Reuters corpus) for coarse-grained semantic disambiguation (corresponding SemEval 2007 task [455])

show that the proposed model performs similar to or above the state of the art, with the authors stressing the reusability of their approach.

In [270], the authors introduce “Sensembed”, a framework that generates both semantic annotation tags for a given dataset and sense-level embeddings from the resulting annotated corpus. In their approach, the BabelNet semantic graph [457] is used to annotate a Wikipedia dump so as to create a large semantically-annotated corpus via the BabelFly WSD algorithm [450]. Out of the disambiguated corpus, sense vectors are produced with the CBOW model [440]. These vectors are subsequently evaluated in multiple word similarity, relatedness and word-context similarity tasks on multiple datasets. SensEmbed outperforms lexical word embeddings, as well as many related sense-based approaches on word similarity, with respect to Spearman correlation. It also performs better than mutual information-based baselines and word2vec embeddings on the SemEval-2012 relational similarity task [297]. SensEmbed is used in [142], where the knowledge base disambiguation and unification framework “KB-UNIFY” employs sense embeddings for the disambiguation stage, along with cross-resource entity linking and alignment, so as to unify and merge semantic resources. In [181], the authors employ WordNet supersenses, *i.e.*, flat groupings of synsets, denoting synset high-level and more abstract semantic information than regular WordNet entities. BabelNet synsets are mapped to WordNet supersenses, using an automatically annotated Wikipedia corpus at multiple abstraction levels [564]. A range of evaluations on downstream classification tasks (subjectivity, metaphor, polarity classification) demonstrates that the proposed approach yields state of the art results, outperforming the exclusive use of distributional information. Further, the “AutoExtend” method [532] uses WordNet, considering words and synsets as a sum of their lexemes. Word embeddings are learned (or existing ones are modified) by a deep autoencoder, with the hidden layers representing synset vectors. Experiments on WSD, using the SemEval task corpora, show that AutoExtend achieves higher accuracy than an SVM-based approach with multiple engineered semantic features, with a subsequent combination of the two approaches further improving performance – indicating complementarity between them. In addition, an evaluation on word similarity shows that AutoExtend outperforms other systems ([262, 102, 440]) as well as synset-level embeddings, in terms of Spearman’s correlation. In [216], semantic embeddings are computed independently: a probabilistic random walk over the semantic graph outputs sequences of synsets, with the latter mapped to words via a dictionary of WordNet gloss relations. The resulting pseudo-corpus is fed to the skip-gram algorithm to learn semantic embeddings. Lexical and semantic vec-

tors are subsequently combined in various ways, with simple concatenation outperforming more sophisticated semantic augmentation methods such as retrofitting, on similarity and relatedness datasets and tasks.

The approach in [502] examines the effect of sense and supersense information on text classification and polarity detection tasks. Disambiguation is performed by mapping the input sentence into a subgraph of the semantic resource containing all semantic candidates per word. Then, the sense with the highest node degree is picked for each word, discarding the rest and pruning the subgraph accordingly. Additionally, supersenses are produced via averaging synset vectors with respect to the grouping of senses provided in WordNet lexicographer files. An experimental evaluation over the BBC, 20-Newsgroups and Ohsumed datasets shows that their approach introduces significant benefits in terms of F1-score, consistently improving the lexical embedding baseline on randomly initialized vectors. However, no improvement is observed when using pre-trained embeddings. For polarity detection, no consistent improvement is reported either. This is attributed to the short document sizes and the lack of word ambiguity in the examined datasets.

These approaches effectively introduce semantic information to deep neural architectures and word embeddings, but the evaluation of the refined embeddings on applied machine learning scenarios is limited, focusing for the most part on a variety of semantic similarity tasks or specialized, domain-specific classification tasks. Instead, this study focuses on a specific machine learning task, namely text classification, exploring the effect of semantic augmentation on deep neural models to the classification performance. Our work is focused on the feature level, applying semantic enrichment on the input space of the classification process. We separate the embedding generation from the semantic enrichment phase, as in [174], where the semantic augmentation can be applied as a post-processing step. In fact, we model the semantic content as a separate representation of the input data that can be combined with a variety of embeddings, features and classifiers in order to enhance their expressive capabilities and improve their explainability. Our approach extends earlier work on shallow features and learners [158, 462, 562, 413] by augmenting deep embedding generators instead of local features. We also expand our investigation to additional semantic extraction and disambiguation approaches, by considering the effect of the n -th degree hypernymy relations and of several context semantic embedding methods. Finally, we expand and complement the findings of [502], adopting multiple disambiguation schemes and a comparatively lower complexity architecture for classification.

5.1.3 Text preprocessing and embedding generation

We begin by applying preprocessing to each document in order to discard noise and superfluous elements that are deemed irrelevant or even harmful for the task at hand. The processing tokenizes the original texts into a list of words and discards non-lexical elements such as punctuation, whitespace and stopwords¹. To generate word embeddings, we employ the established word2vec algorithm [440]. Specifically, we use the Continuous

¹For stopwords removal, we use a popular list from [508].

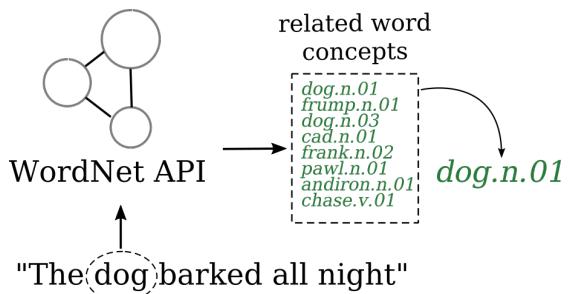


Figure 5.2: Example of the *basic* disambiguation strategy. Given the list of candidate synsets from the NLTK WordNet API, the first item is selected.

Bag-of-Words (CBOW) variant for the training process, which produces word vector representations by modelling co-occurrence statistics of each word based on its surrounding context. Instead of using pre-trained embeddings, we extract them from the given corpus, using a context window size of 10 words. To discard outliers, we also apply a filtering phase, which discards words that fail to appear at least twice in the training data. We train the embedding representation over 50 epochs (*i.e.*, iterations over the corpus), producing 50-dimensional vector representations for each word in the resulting dataset vocabulary. These embeddings represent the textual / lexical information of our classification pipeline.

5.1.4 Semantic enrichment

We now elaborate on the core of our approach, which infuses the trained embeddings with semantic information. First, we describe the semantic resource we use, WordNet. Then we introduce the semantic disambiguation phase which, given a word, selects a single element from a list of WordNet concepts as appropriate for the word. We continue with a description of the propagation mechanism we apply to spread semantic activation, *i.e.* to include more semantic information related to the concept in the word representation. We conclude with the fusion strategy by which we combine all information channels to a single, enriched representation.

5.1.4.1 Semantic resource

We use WordNet [441]², a popular lexical database for the English language that is widely used in classification and clustering tasks [268, 449, 383, 158]. WordNet consists of a graph, where each node is a set of word senses (called synonymous sets or *synsets*) representing the same approximate meaning, with each sense also conveying part-of-speech information.

Synset nodes are connected to neighbours through a variety of relations of lexical and semantic nature (*e.g.*, *is-a* relations like hypernymy and hyponymy, part-of relations such

²See also <https://wordnet.princeton.edu/>

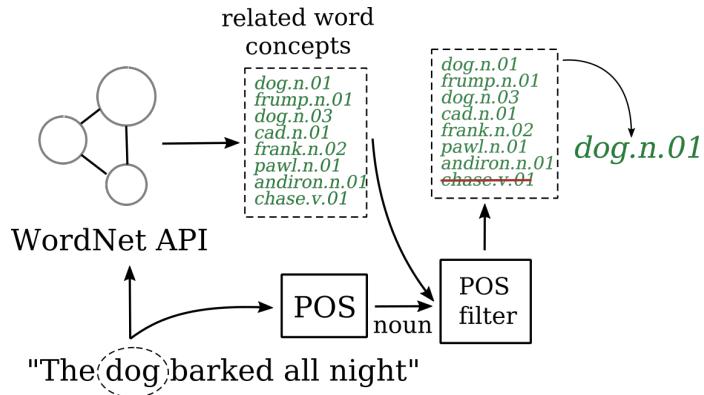


Figure 5.3: Example of the *POS* disambiguation strategy. Given the candidate synsets retrieved from the NLTK WordNet API, the ones that are annotated with a part-of-speech tag that does not match the respective tag of the query word are discarded. After this filtering process, the *basic selection strategy* is applied.

as meronymy, and others). Note that while the terms “synset” and “concept” are similar enough to merit interchangeable use, we will use the word “concept” throughout the study when not talking about the internal mechanics of WordNet.

Concerning WordNet synsets, in the following paragraphs we employ a notation inspired by [454], as follows. In WordNet, the graph node encapsulating the everyday concept of a dog is a synset s , composed of individual word senses of near-identical semantic content, as below:

$$s = \{ \text{dog}_n^1, \text{domestic_dog}_n^1, \text{Canis_familiaris}_n^1 \}$$

The subscript of each word sense denotes POS information (e.g. nouns, in this example), while the superscript denotes a sense numeric identifier, differentiating between individual word senses. Given a word sense, we can unambiguously identify the corresponding synset, enabling us to resolve potential ambiguity [417, 454] of polysemous words. Thus, in the following paragraphs we will use the notation $l.p.i$ to refer to the synset that contains the i -th word sense of the lexicalization l that is of a part-of-speech p .

For example, the common meaning of the word “dog” is approximately aligned with any of the word senses in the synset s , which in WordNet is accompanied by a definition: “*a member of the genus Canis (probably descended from the common wolf that has been domesticated by man since prehistoric times; occurs in many breeds)*”. However, additional, more obscure senses of “dog” can be found in WordNet – e.g. dog_n^3 , mapped to the synset defined as “informal term for a man”³.

Thus, synset s will be referred to as $\text{dog}.n.01$, directly pointing to the (first) word sense of the word noun “dog”⁴. Using this notation, we move on to describe the disambiguation process.

³As in phrases like “*You lucky dog!*”.

⁴This notation is also the one employed by the WordNet interface used in our semantic extraction process, the description of which follows.

5.1.4.2 Disambiguation

We extract information from WordNet via the NLTK interface⁵. Its API supports the retrieval of a collection of synsets as possible semantic candidates for an input word. It also allows traversal of the WordNet graph via the synset relation links mentioned above. Below we will denote string literals with a quoted block of text (e.g. “dog”).

To select the most relevant synset from the acquired response of the API, we employ one of the following three disambiguation strategies:

1. The *basic* disambiguation strategy serves as a baseline, simply selecting the first synset from the retrieved list, discarding the rest. The NLTK WordNet API ranks the retrieved synsets for a word query with respect to corresponding word sense frequencies computed in the SemCor sense-tagged corpus [442, 417, 454]. Therefore, this method selects the most common meaning for the word (as computed in the SemCor text collection). An illustration of this selection process is shown in Figure 5.2.
2. The *POS* disambiguation strategy first filters the retrieved list of candidate synsets. The filtering discards all synsets that do not match the part-of-speech (POS) tag of the query word. Then the first remaining synset is selected.

For example, when the word “can” is supplied with the POS tag “verb”, the synset defined as “airtight sealed metal container for food or drink or paint etc.” is discarded from the candidate synset list. After this filtering phase, the same mechanism as with the *basic* disambiguation process is applied. See Figure 5.3 for a visualization of this process.

3. The *context-embedding* disambiguation strategy uses a semantic embedding approach. For each candidate synset, related words are extracted from the accompanying example sentences together with the corresponding synset definition (*i.e.* the *gloss*).⁶ Given the set of words from both sources, we compute the embedding mapping via the process described in Section 5.1.3. This associates every candidate synset with the set of embeddings of all words in its context.

To arrive at a single vector representation for every candidate synset, this strategy averages all components across word embeddings in the context. This process maps the semantic information into a shared representation with the lexical / textual one. In other words, it projects synsets into the same vector space. As a result, disambiguation by vector comparison is enabled: given a word w , its embedding e_w and a set of synset embeddings $S = \{e_1, \dots, e_{|S|}\}$, we assign w to the synset $s = \underset{s \in S}{\operatorname{argmin}} [dist(e_w, e_s)]$, where e_s is the aggregated synset embedding for synset $s \in S$ and $dist(\cdot, \cdot)$ denotes a vector distance metric.

⁵<https://www.nltk.org/data.html>

⁶For instance, the *example sentence* for the synset *dog* is “*The dog barked all night*”, while its *definition* was quoted in Section 5.1.4.1. Note that we consider synset definition sentences in addition to the WordNet examples, because around 70% of all synsets are associated with a single example.

To ensure adequate word context for generating representative semantic embeddings, we discard all synsets with fewer than 25 context words. The synset vector computation process from the whole WordNet, which is illustrated in Figure 5.4, results in 753 adequately represented synsets. The disambiguation process itself is depicted in Figure 5.5.

Overall, the *context-embedding* disambiguation strategy performs synset selection in a significantly more complicated manner than the other two strategies. Rather than using low-level lexical information (*basic* strategy) or lexical and syntactic features (*POS* strategy), this approach exploits the available distributional information in WordNet in order to match the input word to a synset.

This strategy bears some resemblance to other embedding-based disambiguation methods in the literature. However, given (a) the focus of this investigation on the downstream task of classification and (b) the multiple other disambiguation strategies examined, we decided to build a straightforward approach as described above; this way, we managed to reduce both the number of decision points (thus largely avoiding heuristics engineering and meta-parameter optimization) as well as the computational requirements of this embedding-based disambiguation approach, in favor of a more robust, readily applicable algorithm. Comparatively, an example of a similar embedding-based approach is the work in [102], where the authors build synset embeddings via a process that includes averaging vectors of words that are related to WordNet synsets. However, their method is considerably more intricate compared to ours, since in their approach (i) semantic vectors are additionally fitted, with the aforementioned scheme being used just for sense vector initialization, (ii) a filtering step is used to process the WordNet gloss text prior to vector initialization, using a predefined subset of POS tags and iii) only words with representations close to the candidate word in a embedding space are considered, using a distance / similarity threshold. In contrast, *context-embedding* directly pools all available textual resources that accompany a synset in order to construct an embedding, *i.e.* utilizing all available distributional information WordNet has to offer. Additionally, no further fitting is applied to the resulting embedding, but it is used as-is in the downstream classification task for disambiguation purposes (*i.e.* to retrieve the synset that will be used in the actual semantic information extraction component).

5.1.4.3 n -level hypernymy propagation

Since WordNet represents a graph of interconnected synsets, we can exploit meaningful semantic connections to activate relevant neighbouring synsets among the candidate ones. In fact, our approach propagates activations further than the immediate neighbours of the retrieved candidate synsets, to a multi-step, n -level relation set. This way, a spreading activation step [117] propagates the semantic synset activation towards synsets connected with hypernymy relations with the initial match. In other words, it follows the edges labelled with *is-a* relations to include the encountered synsets in the pool of retrieved synsets.

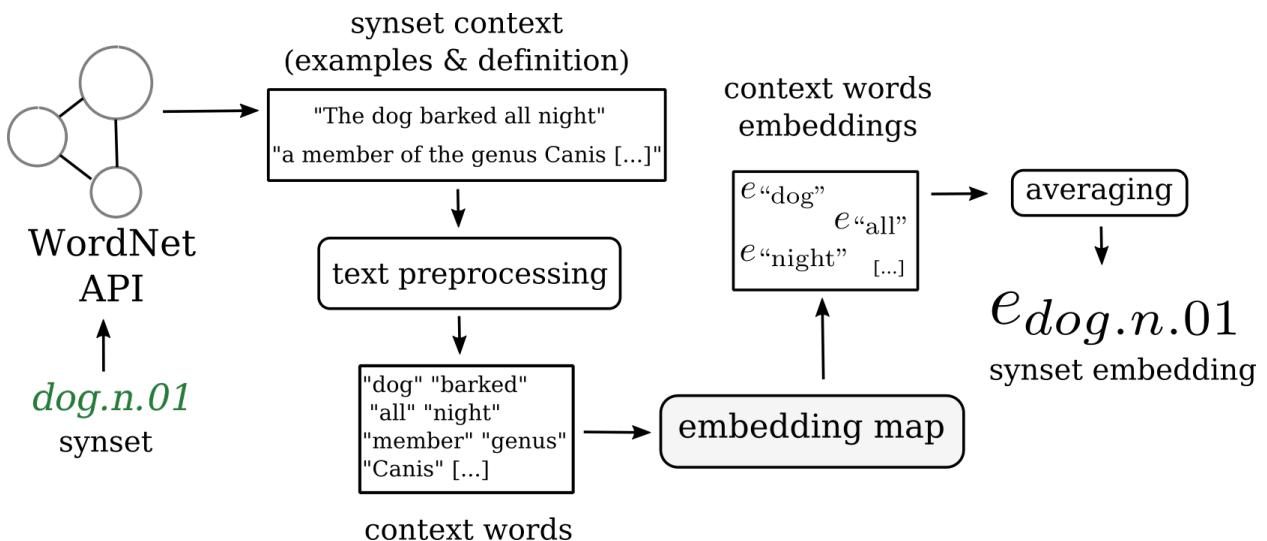


Figure 5.4: Example of the synset vector generation for *context-embedding* disambiguation strategy. The context of each synset is tokenized into words, with each word mapped to a vector representation via the learned embedding matrix. The synset vector is the centroid produced by averaging all context word embeddings.

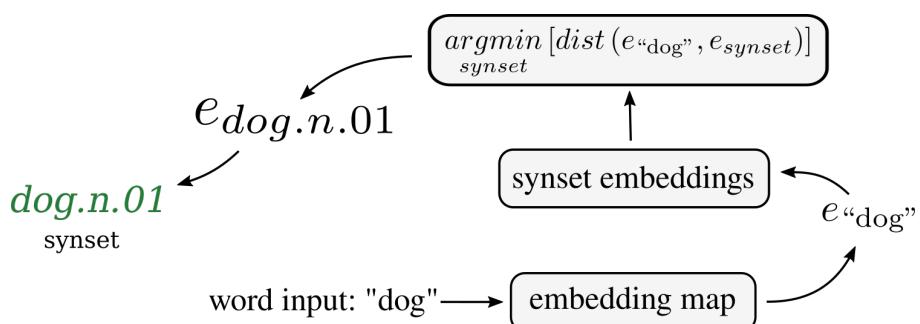


Figure 5.5: Example of the disambiguation phase of the *context-embedding* disambiguation strategy. A candidate word is mapped to its embedding representation and compared to the list of available synset vectors. The synset with the vector representation closest to the word embedding is selected.

The synsets extracted with this process are annotated with weights inversely proportional to the distance of the hypernymy level from the original synset. This weight decay is applied to diminish the contribution of general and/or abstract synsets, which are expected to be encountered frequently, thus saturating the final semantic vector.

After a hyper-parameter tuning phase, we arrived at the configuration of a 3-level propagation process, with each level associated with a weight decay factor of 0.6. This mechanism enables words to share semantic information even if they do not belong to the same synset directly, but their mapped synsets can be linked via a short walk in the semantic graph – the longer the path, the lower the resulting relatedness weight.

To illustrate the spreading activation mechanism, consider querying the NLTK WordNet interface with the input word “dog” as an example. Using the *basic* disambiguation strategy (cf. Section 5.1.4.2), the first synset is selected out of the retrieved list – *i.e.* is the synset $\text{dog.n.01} = \{\text{dog}_n^1, \text{domestic_dog}_n^1, \text{Canis_familiaris}_n^1\}$ and is valued with a unit weight. Subsequently, our spreading activation procedure is activated and operates as follows:

- The first step yields the direct hypernyms of the synset dog.n.01 in the WordNet graph: $h_1 = \{x | \text{dog.n.01 is-a } x\} = \{\text{canine.n.02}, \text{domestic_animal.n.01}\}$, with $\text{canine} = \{\text{canine}_n^2, \text{canid}_n^2\}$ and $\text{domestic_animal.n.01} = \{\text{domestic_animal}_n^1, \text{domesticated_animal}_n^1\}$. These two synsets are thus assigned a weight of $0.6^1 = 0.6$.
- Next, we retrieve the hypernyms of each synset in h_1 . This yields the synsets $h_2 = \{\text{carnivore.n.01}, \text{animal.n.01}\}$, each weighted with $0.6^2 = 0.36$.
- Finally, the third step produces the synsets $h_3 = \{\text{placental.n.01}, \text{organism.n.01}\}$, each with a weight of $0.6^3 = 0.216$.

As a result, the result of the 3-level spreading activation procedure on the word “dog” is a semantic vector with values: [1, 0.6, 0.6, 0.36, 0.36, 0.216, 0.216] corresponding to weights for the synsets [dog.n.01 , canine.n.02 , $\text{domestic_animal.n.01}$, carnivore.n.01 , animal.n.01 , placental.n.01 , organism.n.01]. This example is also illustrated in Figure 5.6. Having this mapping of word to name-value collections, the next section describes the procedure by which the word-level information is fused to arrive at document-level vectors.

5.1.4.4 Fusion

As explained above, each semantic extraction process yields a set of concept-weight pairs for each word in the document. We want a single, constant length, semantic vector for each document. Thus, we form this vector by following a bag-of-synsets/concepts approach: we create a vector space where each dimension is mapped to one of the concepts discovered in the corpus; then we apply the semantic extraction to all documents in the corpus, mapping each of these documents in the space based on the frequency of a concept in the document. Thus, similar to the bag-of-word paradigms, we can generate two different vector types:

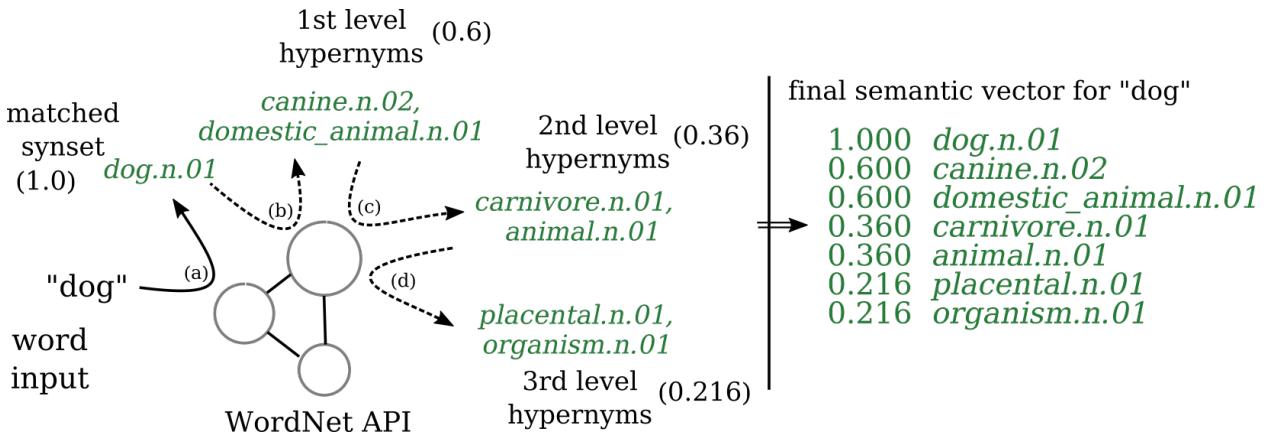


Figure 5.6: Example of the spreading activation process for the input word “dog”, executed for 3 levels with a decay factor of 0.6. The solid line (a) denotes the semantic disambiguation phase with one of the covered strategies. Dashed lines ((b) through (d)) represent the extraction of synsets linked with a hypernymy (*is-a*) relation to any synset in the source list. The numeric values represent the weight associated with synsets of each level, with the final semantic vector for the input word being listed to the right.

1. We consider the raw concept frequencies over each document, arriving at semantic vectors of the form $s^{(i)} = \{s_1, s_2, \dots, s_d\}$, where $s_j^{(i)}$ denotes the frequency of the j -th concept in the i -th document. A concept in the semantic vector appears at least once in the training dataset. Note that concepts extracted from the test dataset, which do not appear in the training dataset are discarded.
2. We apply a weighting scheme similar to TF-IDF [544] at the document and corpus levels, *i.e.*, by normalizing the document-level concept frequencies with the corresponding corpus-level frequencies: $w_j^{(i)} = s_j^{(i)} / \sum_{k \in [0, \dots, N]} s_j^{(k)}$, where $s_j^{(i)}$ stands for the raw frequency of the j -th concept in the i -th document, N is the number of documents in the dataset and $w^{(i)}$ is the final TF-IDF concept weight in the i -th document. This process reduces the importance of concepts that appear in too many documents in the corpus, similar to weight discounting of common words in a text retrieval setting.

After this post-processing stage, we are ready to incorporate the semantic vector into the classification pipeline. This is done in two ways:

1. The *concat* fusion strategy concatenates the embedding with the semantic vector, arriving at a semantically augmented representation that is fed to the downstream classifier.
2. The *replace* fusion strategy discards completely the lexical embedding, using only the semantic information for tackling the categorization task.

Since all cases above use semantic features that represent explicit concept-weight information (rather than explicitly distributed vectors), we do not fine-tune the augmented

embeddings during training but keep the entire representation “frozen” to the original input values.

5.1.5 Training

We use a deep neural network (DNN) with 2 hidden layers, each containing 512 neurons. We arrived at this configuration after fine-tuning these two hyper-parameters through a grid search on a range of values: from 1 to 4 with a step of 1 for the number of hidden layers; values of $\{128, 256 \dots, 2048\}$ for the number of neurons within each hidden layer. We use dropout with a heuristically selected 0.3 drop probability and position it after each dense layer to avoid overfitting. We train the DNN for 50 epochs over the total training data, with a 25-epoch early stopping, which allows the training to end prematurely, if the validation loss does not decrease for 25 consecutive epochs. We apply a 5-fold cross-validation split for training, classifying the input via a softmax layer. The learning rate is initialized to 0.1, applying a reduction schedule of a 0.1 decay factor every 10 epochs on loss stagnation.

5.1.6 Workflow summary

At this point we summarize the complete workflow of our approach to put everything we have described together, under a common view:

1. Preprocessing transforms each document into a sequence of informative words.
2. word2vec word embeddings are learned from scratch on these word sequences.
3. Semantic information for each document is extracted via the NLTK WordNet interface in the form of frequency-based concept vectors. This entails:
 - (a) Concept extraction for each word by one of the disambiguation strategies (*basic*, *POS* or *context-embedding*).
 - (b) 3-level hypernymy activation propagation based on the WordNet graph.
 - (c) Raw-frequency or TF-IDF weighting.
4. The semantic information from Step 3 is combined with the word embeddings from Step 2 through a fusion strategy (*concat* or *replace*).
5. Classification with a DNN classifier.

The above workflow is illustrated in Figure 5.7.

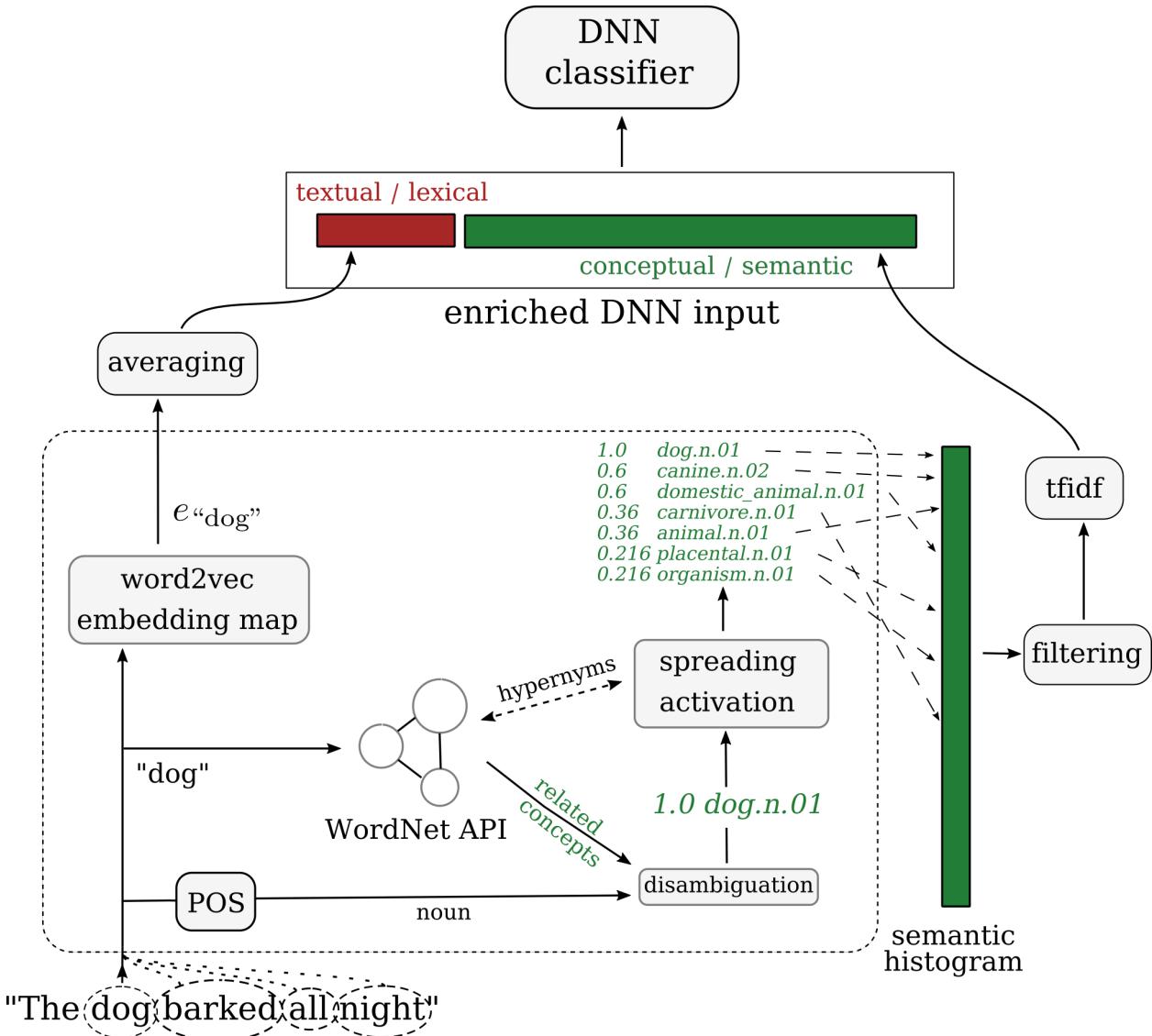


Figure 5.7: An example of the semantic augmentation process, leading up to classification with a DNN classifier. The image depicts the case of *concat fusion*, i.e., the concatenation of the word embedding with the semantic vector. The dashed box is repeated for each word in the document. Green / red colors denote semantic / textual information, respectively.

5.1.7 Experimental evaluation

In this section, we outline the experiments performed to evaluate our semantic augmentation approaches for text classification. In Section 5.1.7.1, we describe the datasets and the experimental setup, in Section 5.1.7.2, we present and discuss the obtained results, and in Section 5.1.8, we compare our approach to related studies.

5.1.7.1 Datasets and experimental setup

We use the 20-Newsgroups dataset [340]⁷, a popular text classification benchmark. This corpus consists of 11,314 and 7,532 training and test instances of user USENET posts, spanning 20 categories (or “newsgroups”) that pertain to different discussion topics (e.g. *alt.atheism*, *sci.space*, *rec.sport.hockey*, *comp.graphics*, etc.). The number of instances per class varies from 377 to 600 for the training set, and from 251 to 399 for the test set, while the mean number of words is 191 and 172 per training and test document, respectively. We use the “bydate” version, in which the train and test samples are separated in time (i.e., the train and the test set instances are posted before and after a specific date).

Additionally, we utilize the Reuters-21578⁸ dataset, which contains news articles that appeared on the Reuters financial newswire in 1987 and are commonly used for text classification evaluation. Using the traditional “ModApte” variant, the corpus comprises 9,584 and 3,744 training and test documents, respectively, with a labelset of 90 classes. The latter correspond to categories related to financial activities, ranging from consumer products and goods (e.g. *grain*, *oilseed*, *palladium*) to more abstract monetary topics (e.g. *money-fx*, *gnp*, *interest*). The dataset is extremely imbalanced, ranging from 1 to 2,877 training instances per class, and from 1 to 1,087 test instances per class. The mean number of words is approximately 92, for both training and test documents. Most instances are labelled with a single class, with few of them having a multi-label annotation (up to 15 labels per instance). While hierarchical relationships exist between the classes, we do not consider them in our evaluation.

Given that we are only interested in single-label classification, we treat the dataset as a single-labelled corpus by using all sample and label combinations that are available in the dataset. This results in a noisy labelling that is typical among folksonomy-based annotation [495]. In such cases lack of annotator agreement occurs regularly and increases the expected discrimination difficulty of the dataset, as we discard neither superfluous labels nor multi-labelled instances.

The technical details of each dataset are summarized in Table 5.1. Apart from sample and word count information, we additionally include: (a) quantities pertaining to the part-of-speech information useful for the POS disambiguation method; (b) the amount of semantic information minable from the text. The POS annotation count and the synset/concept counts are expressed as ratios with respect to the number of words per document.

⁷<http://qwone.com/~jason/20Newsgroups/>

⁸<http://www.daviddlewis.com/resources/testcollections/reuters21578/>

	20-Newsgroups		Reuters	
attribute	train	test	train	test
samples	11,314	7,532	9,584	3,744
class samples	377 - 600	251 - 399	1 - 2,877	1 - 1,087
words	191.164 (587.7)	172.196 (471.37)	92.532 (92.03)	92.899 (105.25)
POS	0.716 (0.07)	0.713 (0.06)	0.672 (0.10)	0.669 (0.10)
WordNet	0.572 (0.09)	0.566 (0.09)	1.479 (0.37)	1.381 (0.38)

Table 5.1: Technical characteristics of the 20-Newsgroups and Reuters datasets. Class samples refers to the range of the number of instances per class, while the last three rows report mean values, with the corresponding standard deviation in parenthesis. The values in the last two rows (POS, WordNet) are expressed as ratios with respect to the number of words per document.

We note that no contextual or domain-specific information is employed in our experiments; reflected by the broad task handled in this work (*i.e.* text classification), both datasets are handled in an identical manner by every configuration examined in the experimental evaluation in the following section. This decision introduces an applicability / performance trade-off: first, the lack of special treatment enables each configuration to be readily applicable to any dataset, with the experimental evaluation better reflecting the generalization capability of the approach. This translates to a single processing pipeline for all datasets, without over-engineered solutions on case-specific details. On the other hand, such a strategy may sacrifice performance gains attainable via accommodating domain or dataset-specific issues. However, we believe that demonstrating the generalization ability of the process is more important and, thus, we focus on this aspect in the experiments.

We used python with Keras⁹ [109] and TensorFlow¹⁰ [2] to build the neural models. All experiments are reproducible via the code that is available on GitHub¹¹. Document pre-processing was performed with Keras and NLTK¹² [394]. We use WordNet version 3.0 for semantic information extraction via the interface available in NLTK¹³. The datasets and semantic resources were acquired via the scikit-learn¹⁴ and NLTK APIs¹⁵.

5.1.7.2 Results

We now present the results of our experimental evaluation, discussing the performance of each method per dataset.

In the following tables, we present results in accuracy and macro F1-score (columns “ac-

⁹<https://keras.io/>

¹⁰<https://www.tensorflow.org/>

¹¹<https://github.com/npit/nlp-semantic-augmentation/tree/jnle>

¹²<https://www.nltk.org/>

¹³<http://www.nltk.org/howto/wordnet.html>

¹⁴<https://scikit-learn.org/stable/datasets/index.html>

¹⁵<https://www.nltk.org/book/ch02.html>

curacy” and “ma-f1”, respectively), in terms of mean values over 5 folds. We omit standard deviation scores in favor of compactness and since they consistently fall below 0.005. In the “enrichment” column, the concatenation of the embedding and the semantic vector is denoted by “concat”, whereas “replace” indicates the replacement of the former with the latter. The “features” column reports the use of raw concept frequencies (“freq”) or TF-IDF weights (“tfidf”). The “disam” column indicates the disambiguation strategy, i.e. “basic”, “POS” and “context” corresponding to *basic*, *POS content-embedding* respectively. The “+spread” suffix denotes the use of the spreading activation that is outlined in Section 5.1.4.2. Finally, the dimensionality of each data vector is reported in the “dim” column. All results are obtained by training and evaluating the DNN model that is described in Section 5.1.5. Note that we include two baseline methods: the first row corresponds to the *majority classifier*, which always selects the class with the most samples in the training dataset, while the second row corresponds to word2vec embeddings, without any semantic augmentation applied in the vector (“embedding-only”).

In Table 5.2, we present the experimental results for the 20-Newsgroups dataset, which give rise to the following observations:

- The introduction of semantic information brings gains in both accuracy and macro F1-score. The best combination concatenates raw frequency-based concept vectors to the word embeddings, with disambiguation applied according to the *basic* selection strategy.
- Regarding the concept selection method, projecting semantic vectors in the embedding vector space does not improve performance. Part-of-speech filtering yields marginally inferior results than selecting the first retrieved synset from the WordNet API, which is the simplest and best-performing approach.
- The 3-rd order hypernymy propagation via the spreading activation mechanism does not improve the baseline semantic augmentation in a consistent manner.
- Concatenating the word embedding with the semantic vector consistently outperforms the replacement of the former with the latter to a large extent.
- Raw concept frequencies always outperform the TF-IDF normalized weights.

Given these results and observations, we move on to an error analysis of our system by examining the performance of our best-performing configuration in more detail. To this end, Figure 5.8(a) illustrates the classification error via the confusion matrix for the best-performing configuration. To aid visualization, the diagonal has been removed. We observe that misclassification is approximately fairly concentrated in the first 6 classes¹⁶ (i.e., *alt.atheism*, *comp.graphics*, *comp.os.ms-windows.misc*, *comp.sys.ibm.pc.hardware*, *comp.sys.mac.hardware* and *comp.windows.x*), with peaks appearing at classes 15 and 16 (*soc.religion.christian* and *talk.politics.guns*, respectively). Additionally, Figure 5.8(b) depicts the label-wise performance for the best-performing configuration. We can see that

¹⁶For a complete reference regarding the mapping of numeric indexes to class names, refer to Table A.1 in the appendix.

most labels perform at an F1-score above a value of 0.6, with class 10 (*rec.sport.hockey*) being the easiest to handle by our classifier and class 19 (*talk.religion.misc*) being the most difficult.

Furthermore, Table 5.3 presents indicative misclassification cases selected from the erroneous prediction of our best-performing configuration. A number of patterns and explanations in these errors are identified by a manual analysis of the results, hereby outlined by selected examples. Specifically, four cases are identified ((a) through (d)). Example instances for each case are referred to by an ID (e.g. a1, a2, b1, etc.). For each instance we illustrate the true label, the wrong prediction made by our system, and indicative segments found in the instance text.

- First, our system often labels instances with very similar / plausible alternative annotations to the ground truth, which could be however arguably regarded as semantically valid – even by human evaluators – given the instance text. This case is presented in the table group (a) - Semantically similar labels. In example a1, the label *alt.atheism* is applied instead of the true label *talk.religion.misc*, on a 20-Newsgroups instance discussing theism and the conversion of believers to/from atheism, while example a3 is misclassified to *comp.sys.mac.hardware* rather than *sci.electronics*, with the discussion in the text dealing with several computer parts.
- Similarly, our system deviates from the ground truth due to ambiguity and multiple distinct thematic topics in the textual content of many instances. These texts thus approach a multi-label nature with respect to the available classes. This case is reflected in group (b) – Ambiguous / equivocal instances – where we list discussions that involve multiple terms and keywords connected to many classes. For example, b1 contains references to hardware but has a ground truth generally related to the Windows operating system, while b2 mentions multiple graphics file formats but the true label is the X window system. Instances b3 and b4 are mislabelled as sales-related posts from related keywords and terms (e.g. “revenue”, “business”, “for sale”, etc.), instead of classes linked to specific products and objects of discussion. Finally, text b5 features a lengthy discussion on US abortion legislation which was labelled as a political posting by our system, rather than a religious one.
- Further, there are cases where the content of the text is critically linked to a single or very few mentions of a named entity, that our model either disregards or the available data is not sufficient to leverage (table section (c) - Critical named entity). Such cases are illustrated with examples c1 and c2, where knowledge that “Jack Morris” is a baseball player or that “VAX” is an IBM machine would be required to reach the correct conclusions.
- Additionally, some errors can arise due to the disambiguation method failing to produce to the correct sense, given the context. These cases are presented in group (d) - Context miss. In instance d1, the system associates mentions of battery features to description of product aspects and/or bad reviews, predicting a sales-related text. In example d2, our model gives a very large weight on “The Devil Reincarnate” user handle, deducing a religion class from that very mention and undervaluing the other

enrichment	features	disam	accuracy	ma-f1	dim
N/A	majority-base	N/A	0.005	0.053	N/A
N/A	embedding-only	N/A	0.724	0.716	50
concat	freq	basic	0.784	0.790	18,157
concat	tfidf	basic	<u>0.752</u>	<u>0.760</u>	18,157
replace	freq	basic	<u>0.763</u>	<u>0.769</u>	18,107
replace	tfidf	basic	0.643	0.648	18,107
concat	freq	pos	<u>0.778</u>	<u>0.784</u>	19,549
concat	tfidf	pos	<u>0.755</u>	<u>0.762</u>	19,549
replace	freq	pos	<u>0.764</u>	<u>0.770</u>	19,499
replace	tfidf	pos	0.648	0.652	19,499
concat	freq	basic+spread	<u>0.775</u>	<u>0.782</u>	22,116
concat	tfidf	basic+spread	<u>0.755</u>	<u>0.762</u>	22,116
replace	freq	basic+spread	<u>0.761</u>	<u>0.768</u>	22,066
replace	tfidf	basic+spread	0.652	0.656	22,066
concat	freq	pos+spread	<u>0.778</u>	<u>0.784</u>	23,546
concat	tfidf	pos+spread	<u>0.762</u>	<u>0.755</u>	23,546
replace	freq	pos+spread	<u>0.768</u>	<u>0.762</u>	23,496
replace	tfidf	pos+spread	<u>0.766</u>	<u>0.760</u>	23,496
concat	tfidf	context	0.714	<u>0.722</u>	803
concat	freq	context	0.719	<u>0.725</u>	803
replace	tfidf	context	0.563	0.569	753
replace	freq	context	0.673	0.679	753

Table 5.2: 20-Newsgroups main experimental results. Underlined values outperform the “embedding-only” baseline method, while **bold** values indicate the best dataset-wise performance. Values in *italics* represent a performance boost achieved by the spreading activation in comparison to the identical configuration without it. “N/A” stands for non-applicable.

text terms. In d3, mentions of “disk” are likely linked to the hard disk drive mechanical component, rather than the software-centric sense in the context of operating systems, leading to a corresponding misclassification.

Cases of classification error not included below may be harder to explain; potential causes for them could involve data outliers, classifier bias due to sample / instance size imbalances, etc.

Having performed an error analysis of our model, we assess the statistical significance of the performance improvements introduced by each semantic enrichment configuration. To this end, we present in Table 5.4 the pairwise t-test results of the experimental configurations presented above, with respect to the “embedding-only” baseline method. We can see that all configurations achieve significantly different results at a 5% confidence level, with most configurations also performing even more consistently differently, at a 1% confidence level. Regarding the statistical significance of hypernymy propagation, we assess

id	true / predicted label	indicative text
(a) - Semantically similar labels		
a1	talk.religion.misc / alt.atheism	scriptures are the whole truth / convert non-theism / religion is a nice fantasy
a2	talk.religion.christian / talk.religion.misc	evangelical counter-cult organization / World of Faith / historic Christian faith
a3	comp.sys.mac.hardware / sci.electronics	Intel processor / HP-IB connector / Mac board NATIONAL INSTRUMENTS
(b) - Ambiguous / equivocal instances		
b1	comp.os.ms-windows.misc / comp.sys.ibm.pc.hardware	McAFree anti-virus program / scan entire hard disk / my friend's machine
b2	comp.windows.x / comp.graphics	converters for xpm
b3	sci.electronics / misc.forsale	converting GIFs/JPEGs/PS / xpm format
b4	rec.autos / misc.forsale	lost revenue due to pirates / business environment / purchase a few copies
b5	talk.religion.misc / talk.politics.misc	MR2 seats for sale / gave the seller / ask questions when buying
(c) - Critical named entity		
c1	rec.sport.baseball / rec.autos	Tieing Abortion to Health Reform / Clinton
c2	comp.sys.ibm.pc.hardware / comp.sys.mac.hardware	cooling fan / heat sink grease
(d) - Context miss		
d1	sci.electronics / misc.forsale	this Jack Morris fella / season's only
d2	rec.autos / talk.religion.misc	just started / how Morris is doing
d3	comp.os.ms-windows.misc / comp.sys.ibm.pc.hardware	VAX/VMS VNEWS / PC Power
		cooling fan / heat sink grease
		Lead Acid batteries / battery
		goes dead / plates of the battery
		The Devil Reincarnate / commercial
		cars today / VW Golf/Passat
		disk copy / PC has 4MB
		RAM / allocate more memory

Table 5.3: Misclassification cases for the best-performing configuration over the 20-Newsgroups dataset, where (a) the predicted label is semantically similar to the ground truth, (b) the test instance can be reasonably considered semantically ambiguous, given the labelset, (c) the error is related to the existence of critical named entities, or (d) the error is linked to context misidentification. True / predicted labels refer to the instance ground truth and the erroneous prediction of our model for that test instance, respectively. The listed slash-separated text segments from each instance are indicative samples believed to have had a contribution to misclassification.

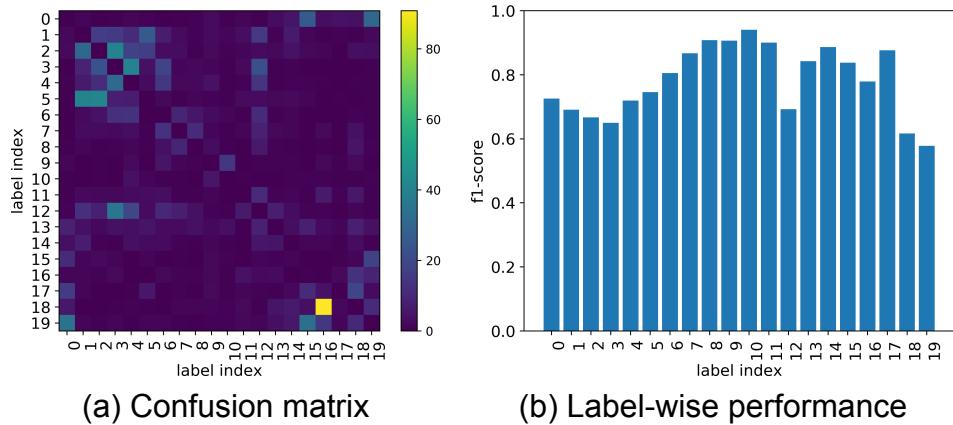


Figure 5.8: (a) The diagonal-omitted confusion matrix, and (b) the label-wise performance chart for our best performing configuration over the 20-Newsgroups dataset.

it by comparing its presence and absence, *i.e.*, “X+spread” rows against “X” rows. Min / max p values ranged from $9.97e-06$ to $7.50e-01$ for the accuracy score and from $1.29e-05$ to $9.78e-01$ for the macro F1-score (due to lack of space, we omit detailed results). There, statistical significance was achieved at the 5% confidence level for the majority of configurations, with many configurations (including those composed of *POS* concept selection, *replace* fusion and TF-IDF weights) surpassing the 1% confidence level for accuracy.

Next, we examine the effect of dimensionality reduction on the performance of our semantic vectors. Table 5.5 reports the performance when discarding concepts with a raw concept-wise frequency of at least 20 occurrences, while Table 5.6 corresponds to keeping only the top-50 concepts, in terms of dataset-wise frequency, so as to match the word embedding dimension. We apply both filtering techniques on all configurations of our approach, except for hypernymy propagation and context embedding, due to space limitations. We observe the following patterns:

- The frequency threshold of 20 reduces the dimensionality of the semantic vectors by more than 50%, at a minor cost in classification accuracy. Still, all configurations – excluding TF-IDF with word embedding replacement – surpass the “embedding-only” baseline. In fact, the best configuration of the main experiments (*i.e.*, concatenation fusion with *basic-selected* raw frequencies) maintains a performance very close to its original one.
- When keeping the 50 most-frequent concepts dataset-wise, only concatenation fusion is comparable to the baseline scores. For these cases, TF-IDF weights perform better than the raw concept frequencies, being very close to the baseline results. However, no configuration surpasses the baseline scores.

Table 5.8 reports the experimental results over the Reuters-21578 dataset, which give rise to the following conclusions:

- We observe the same performance patterns as in the case of the 20-Newsgroups

enrichment	features	disam	accuracy	ma-f1
concat	freq	basic	5.92e-06**	4.08e-06**
concat	tfidf	basic	2.54e-05**	6.08e-06**
replace	freq	basic	3.21e-04**	8.24e-05**
replace	tfidf	basic	4.99e-07**	3.62e-07**
concat	freq	pos	4.89e-05**	2.13e-05**
concat	tfidf	pos	8.65e-06**	2.68e-06**
replace	freq	pos	3.48e-04**	1.11e-04**
replace	tfidf	pos	6.73e-06**	9.85e-06**
concat	freq	basic+spread	3.23e-07**	7.51e-08**
concat	tfidf	basic+spread	5.44e-05**	1.88e-05**
replace	freq	basic+spread	7.21e-05**	1.85e-05**
replace	tfidf	basic+spread	4.60e-06**	8.47e-06**
concat	freq	pos+spread	1.73e-04**	8.66e-05**
concat	tfidf	pos+spread	1.84e-05**	2.65e-06**
replace	freq	pos+spread	4.40e-04**	7.65e-05**
replace	tfidf	pos+spread	2.17e-04**	5.09e-05**
concat	tfidf	context	2.51e-03**	1.72e-02*
concat	freq	context	2.95e-02*	2.86e-03**
replace	tfidf	context	3.52e-05**	8.85e-07**
replace	freq	context	3.52e-05**	8.21e-05**

Table 5.4: 20-Newsgroups main experimental pairwise t-test results, with respect to the “embedding-only” baseline. Single and double-starred values represent statistical significance at 5% and 1% confidence levels, respectively.

enrichment	weights	disam	accuracy	ma-f1	dim
N/A	majority-base	N/A	0.005	0.053	N/A
N/A	embedding-only	N/A	0.724	0.716	50
concat	freq	basic	<u>0.779</u>	<u>0.786</u>	7,092
concat	tfidf	basic	<u>0.754</u>	<u>0.761</u>	7,092
replace	freq	basic	<u>0.764</u>	<u>0.771</u>	7,042
replace	tfidf	basic	0.674	0.679	7,042
concat	freq	pos	<u>0.777</u>	<u>0.784</u>	7,323
concat	tfidf	pos	<u>0.752</u>	<u>0.759</u>	7,323
replace	freq	pos	<u>0.763</u>	<u>0.770</u>	7,273
replace	tfidf	pos	0.673	0.678	7,273

Table 5.5: Experiments over the 20-Newsgroups dataset for a concept-wise frequency threshold of 20. Underlined values outperform the “embedding-only” baseline.

enrichment	weights	disam	accuracy	ma-f1	dim
N/A	majority-base	N/A	0.005	0.053	N/A
N/A	embedding-only	N/A	0.724	0.716	50
concat	freq	basic	0.705	0.697	100
concat	tfidf	basic	0.723	0.715	100
replace	freq	basic	0.221	0.212	50
replace	tfidf	basic	0.128	0.081	50
concat	freq	pos	0.705	0.697	100
concat	tfidf	pos	0.722	0.714	100
replace	freq	pos	0.219	0.209	50
replace	tfidf	pos	0.126	0.079	50

Table 5.6: Experiments over the 20-Newsgroups dataset for a dataset-wise frequency threshold of 50. No configuration outperforms the “embedding-only” baseline.

dataset. The semantic augmentation outperforms the “embedding-only” baseline, but not in all cases. The feature vectors are again high-dimensional, although considerably shorter than the 20-Newsgroups dataset. This should be attributed to the fewer training documents and the significantly shorter documents (in terms of words) in the Reuters dataset.

- The best performance is obtained when using raw frequency features, concatenated to the word embedding with part-of-speech selection, in combination with hypernymy propagation via spreading activation. This configuration gives the best accuracy (0.749), closely followed by the same configuration with *basic* selection. For the macro F1-score, replacing the word embedding with frequency-based vectors, part-of-speech selection and hypernymy propagation performs the best (0.378). In fact, the replace fusion strategy holds the top 3 configurations with respect to macro F1-score.
- Regarding concept selection, the *basic* and the POS techniques result in very similar performance. Context embedding selection exhibits poor accuracy, performing under the “embedding-only” baseline, save for marginal improvements in macro F1-score.
- Spreading activation performs the best here.
- Comparing the replacement fusion strategy with the concatenation one almost always favors the latter, with considerable performance difference.
- The TF-IDF weights always perform under the raw concept frequencies, as in the 20-Newsgroups case.

Similarly to the 20-Newsgroups dataset case, we move on to the error analysis, with Figure 5.9(a) depicting the confusion matrix with the misclassified instances (*i.e.* diagonal entries are omitted). For better visualization, it illustrates only the 26 classes with at least 20 samples, due to the large number of classes in the Reuters dataset. We observe that

the misclassification occurrences are more frequent, but less intense than those in the 20-Newsgroup dataset. Noticeable peaks are in classes *crude*, *grain* and *money-fx*. Additionally, Figure 5.9(b) depicts the label-wise performance of our best configuration. We observe that it varies significantly, with classes like *earn* and *acq* achieving excellent performance, while others perform rather poorly, e.g., *soybean* and *rice*.

Moving on to a manual inspection of misclassified instances from the Reuters test set, Table 5.7 presents such examples, produced when using our best-performing configuration. As in the previous section, we pair the listed examples with the true and predicted labels, indicative terms in the text and possible explanations for the erroneous result, given the instance content:

- Firstly, we observe that Reuters includes labels with considerable semantic similarity to others, presented in the first table group ((a) - Semantically similar labels). For example, the class *coconut* is close to *coconut-oil*, while labels like *grain*, *rye* and *wheat* cover varying levels of specificity among types of plant grains and crops. Instance a1 includes coconut production-related terms, while a2 contains multiple mentions of wheat and its pricing and a3 details production information of a variety of grain crops. Likewise, generic and specific classes for seed crops (e.g. *oilseed*, *rapeseed*) as well as vegetable seed oils (i.e. *rape-oil*, *veg-oil*) can be considered as exhibiting semantic overlap. For example, instance a5 is misclassified as such, with its text containing multiple references to various kinds of vegetable oils. For these examples, mislabelling is manifested from the generic ground truth to a deviation to a more specific class, or vice-versa.
- Secondly, scenarios where polysemous instances are estimated to contribute to misclassification are covered in the error category (b) - Ambiguous / equivocal instances. There, we can find test documents with, e.g., the aluminum ground truth class (*alum*) being mislabelled to *gold* and *yen*, with mentions to the precious metal and the Japanese currency in instances b1 and b2 respectively being a core, non-trivial theme in the text. Similarly, multiple themes can be identified in examples b3 (cocoa and information detailing its national production), b4 (the dollar and foreign monetary exchange, with multiple terms pertaining to the latter) and b5 (shipping and transport of wheat).

Moving on to significance testing, we examine the performance difference between each semantic augmentation configuration and the “embedding-only” baseline, reporting in Table 5.9 the corresponding pairwise t-test results. We observe that most configurations achieve significantly different performance at a 5% confidence level, while all hypernymy propagation runs perform even more consistently better, at a 1% confidence level. Runs with *context-embedding* disambiguation and *concat* fusion do not perform significantly different than the baseline at the examined confidence levels. Regarding the significance of the hypernymy propagation runs with respect to the semantic runs without it, the improvements introduced by all configurations of the former are significant at a 1% confidence level: the *p* values range from 3.64e–08 to 1.37e–06 for accuracy, and from 8.54e–08 to 2.57e–06 for macro F1-score (we omit detailed results due to lack of space).

id	true / predicted label	indicative text
(a) - Semantically similar labels		
a1	coconut-oil / coconut	Philippine Coconut Authority / five-year coconut production cycle / coconut product export
a2	grain / wheat	tonnes of soft wheat / USDA approval for wheat price / Continental Grain Co
a3	rye / grain	Danish crops two weeks behind / barley [...] rapeseed / winter wheat, [...] winter rye
a4	oilseed / rapeseed	4,000 tonnes canadian rapeseed / Canadian rapeseed overnight / at an undisclosed price
a5	rape-oil / veg-oil	rise in edible oil demand / soybean oil [...] rapeseed oil / other origin oils
(b) - Ambiguous / equivocal instances		
b1	alum / gold	gold, silver, copper and aluminum / Gold futures which previously had a limit of / the metals market
b2	alum / yen	swapping 32 billion yen / capitalisation to 147 billion yen / Japanese shareholders
b3	cocoa / gnp	German grindings expected to [...] / Grindings rose to 55,190 tonnes / compared to [...] European countries
b4	dlr / money-fx	dilemma over monetary policy / further appreciation of the mark / a weaker dollar would be risky
b5	ship / grain	ship prepares to load wheat / urgently needed wheat for Fiji / Australian Wheat Board spokesman

Table 5.7: Misclassification cases for the best-performing configuration over the Reuters dataset, where (a) the predicted label is semantically similar to the ground truth, or (b) the test instance can be reasonably considered semantically ambiguous, given the labelset. True / predicted labels refer to the instance ground truth and the erroneous prediction of our model for that test instance, respectively. The listed slash-separated text segments from each instance are indicative samples believed to have had a contribution to misclassification.

enrichment	weights	disam	accuracy	ma-f1	dim
N/A	majority-base	N/A	0.2903	0.005	N/A
N/A	embedding-only	N/A	0.725	0.295	50
concat	freq	basic	<u>0.747</u>	0.359	8,534
concat	tfidf	basic	0.723	<u>0.320</u>	8,534
replace	freq	basic	<u>0.744</u>	<u>0.362</u>	8,484
replace	tfidf	basic	0.628	0.222	8,484
concat	freq	pos	<u>0.746</u>	0.349	9,135
concat	tfidf	pos	0.723	<u>0.318</u>	9,135
replace	freq	pos	<u>0.744</u>	<u>0.372</u>	9,085
replace	tfidf	pos	0.627	0.230	9,085
concat	freq	basic+spread	<u>0.748</u>	0.364	11,358
concat	tfidf	basic+spread	0.720	<u>0.322</u>	11,358
replace	freq	basic+spread	<u>0.745</u>	<u>0.376</u>	11,308
replace	tfidf	basic+spread	0.643	0.232	11,308
concat	freq	pos+spread	0.749	0.365	11,966
concat	tfidf	pos+spread	0.720	<u>0.318</u>	11,966
replace	freq	pos+spread	<u>0.746</u>	0.378	11,916
replace	tfidf	pos+spread	0.644	0.243	11,916
concat	tfidf	context	<u>0.726</u>	0.301	803
concat	freq	context	0.724	0.291	803
replace	tfidf	context	0.621	0.147	753
replace	freq	context	0.706	0.277	753

Table 5.8: Reuters main experimental results. Underlined values outperform the “embedding-only” baseline, while **bold** values indicate the best dataset-wise performance. Values in *italics* denote a performance boost by the spreading activation, with respect to the identical configuration without it.

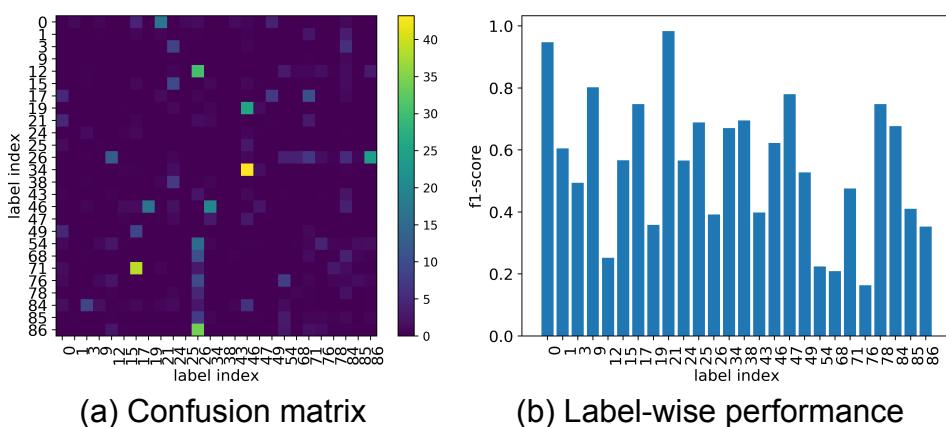


Figure 5.9: (a) The diagonal-omitted confusion matrix, and (b) the label-wise performance chart for our best performing configuration over the Reuters dataset. For better visualization, only the 26 classes with at least 20 samples are illustrated.

enrichment	weights	disam	accuracy	ma-f1
concat	freq	basic	1.80e-04**	1.91e-04**
concat	tfidf	basic	1.29e-01	4.33e-03**
replace	freq	basic	1.04e-04**	2.00e-04**
replace	tfidf	basic	1.32e-06**	7.21e-05**
concat	freq	pos	3.11e-05**	2.40e-03**
concat	tfidf	pos	3.72e-01	1.09e-02*
replace	freq	pos	2.06e-05**	1.17e-04**
replace	tfidf	pos	2.93e-07**	8.84e-04**
concat	freq	basic+spread	2.30e-06**	6.25e-10**
concat	tfidf	basic+spread	3.52e-07**	5.78e-10**
replace	freq	basic+spread	4.41e-07**	5.10e-10**
replace	tfidf	basic+spread	6.94e-08**	1.03e-09**
concat	freq	pos+spread	1.94e-07**	1.83e-11**
concat	tfidf	pos+spread	4.29e-07**	4.21e-10**
replace	freq	pos+spread	1.00e-06**	2.66e-11**
replace	tfidf	pos+spread	1.61e-07**	9.09e-10**
concat	tfidf	context	4.16e-01	2.74e-01
concat	freq	context	2.56e-01	5.03e-01
replace	tfidf	context	1.99e-07**	2.51e-07**
replace	freq	context	3.49e-05**	3.00e-03**

Table 5.9:]

Reuters main experimental pairwise t-test results, with respect to the “embedding-only” baseline. Single and double-starred values represent statistical significance at 5% and 1% confidence levels, respectively.

enrichment	weights	disam	accuracy	ma-f1	dim
N/A	majority-base	N/A	0.2903	0.005	N/A
N/A	embedding-only	N/A	0.725	0.295	50
concat	freq	basic	<u>0.747</u>	<u>0.358</u>	3,286
concat	tfidf	basic	<u>0.726</u>	<u>0.309</u>	3,286
replace	freq	basic	<u>0.745</u>	<u>0.356</u>	3,236
replace	tfidf	basic	0.651	0.231	3,236
concat	freq	pos	<u>0.748</u>	<u>0.363</u>	3,308
concat	tfidf	pos	<u>0.726</u>	<u>0.319</u>	3,308
replace	freq	pos	<u>0.745</u>	<u>0.362</u>	3,258
replace	tfidf	pos	0.653	0.241	3,258

Table 5.10: Experiments over the Reuters dataset for a concept-wise frequency threshold of 20. Underlined values outperform the “embedding-only” baseline.

enrichment	weights	disam	accuracy	ma-f1	dim
N/A	majority-base	N/A	0.2903	0.005	N/A
N/A	embedding-only	N/A	0.725	0.295	50
concat	freq	basic	0.718	0.277	100
concat	tfidf	basic	0.724	0.293	100
replace	freq	basic	0.635	0.155	50
replace	tfidf	basic	0.577	0.045	50
concat	freq	pos	0.717	0.275	100
concat	tfidf	pos	0.725	0.290	100
replace	freq	pos	0.637	0.154	50
replace	tfidf	pos	0.581	0.048	50

Table 5.11: Experiments over the Reuters dataset for a dataset-wise frequency threshold of 50. No configuration outperforms the “embedding-only” baseline.

Finally, applying the dimensionality reduction methods to the semantic vectors yields the results reported in Tables 5.10 and 5.11, which can be summarized as follows:

- The trade-off introduced by the concept-wise frequency threshold of 20 is similar to the 20-Newsgroups case: the dimensionality of the semantic vectors is radically reduced, at a negligible cost in classification performance.
- The combination of TF-IDF weights with the replace fusion strategy exhibits the worst performance, underperforming the “embedding-only” baseline, contrary to the rest of the configurations.
- The 50 most-frequent concepts filtering reduces performance even further, with no configuration surpassing the “embedding-only” baseline. Similar to the 20-Newsgroups case, the combination of the concatenation fusion strategy with the TF-IDF weights performs the best, matching the macro F1-score of the baseline run.

5.1.8 Discussion

We now interpret the experimental findings in relation to the research questions posed in Section 5.1.1 and compare our approach with the state-of-the-art in the field.

5.1.8.1 Addressing the research questions

In light of the experimental results, we revisit the research questions stated in the introduction.

1. **Can semantic information increase the task performance, when applied in this setting? If so, how much?** Experimental results show that a considerable performance boost is achieved by injecting semantic information in the network input, with the improvement achieving significance at a 5% confidence level. Our semantic augmentation approach, which inserts WordNet concept statistics into the network input, achieves the best average performance when using raw concept frequencies, selecting the first retrieved concept per word (*basic* strategy) and concatenating the resulting vector to the word2vec embedding.
2. **Do different semantic disambiguation methods affect the above performance increase and how?** Out of the three examined methods (*i.e.*, *basic*, *POS* and *context-embedding*), the first one seems to suffice, the second one appears to have a non-noteworthy effect on the final performance, while the third one performs under the other two across all datasets. We note however, that the performance of *context-embedding* disambiguation depends heavily on the availability of concept-wise context in the semantic resource. Given that WordNet has limited examples per concept (*e.g.*, 70% of the concepts convey a single example sentence), more credible findings should be derived from a wider investigation that includes semantic resources with a richer lexical content. Alternatively, we could relax the context acquisition constraint, *i.e.*, the minimum word count threshold in Section 5.1.4.2.
3. **What is the effect of enriching the representation with the n -th order hypernymy relations (*e.g.*, through a spreading activation process)?** Initial findings indicate that this effect varies across datasets. For 20-Newsgroups, the hypernymy propagation introduces minor and inconsistent (but statistically significant, at a 5% level) performance boosts, when compared to not using the propagation mechanism. For Reuters, though, the hypernymy propagation gives the best results among the examined configurations, with even greater statistical significance (beyond a 1% confidence level). This discrepancy is probably related to the size of each dataset and to the number of concepts that are extracted from WordNet (the average number of words per document in 20-Newsgroups is double than that in Reuters). Spreading activation apparently does little performance-wise for semantic vectors that are over-saturated, due to the large number of activations that are triggered by the plethora of document words. In fact, spreading activation can be detrimental to performance, if

it introduces additional noise by populating the concept histogram with many generic concepts. As a result, further investigation is required in order to draw more solid conclusions regarding the application of this step. Ideally, that investigation should include semantic filtering steps that are applied after hypernymy propagation.

Next, we summarize the main findings obtained from our experimental analysis:

- **Semantic vector dimensionality:** The semantic vectors of our approach are high-dimensional, a common byproduct of frequency-based features. For 20-Newsgroups, applying dimensionality reduction with a static concept-wise frequency threshold set to 20, allows for restricting the vector length by 61%, while retaining 99.36% of the top accuracy score (Table 5.5). For Reuters, the optimal performance (excluding the hypernymy propagation) stays the same at a 61.5% size reduction (Table 5.10). On the other hand, dimensionality reduction with the 50 most frequent concepts is evidently too extreme, and the resulting fixed-length vectors cannot be used as substitutes to equal-length word embeddings. This is in line with the next observation, which states that replacing the embedding vector entirely, rather than augmenting it, consistently deteriorates the classification performance, at least for the datasets and semantic resources we consider. Notably, in this dimensionality reduction scenario – and contrary to the results of both the main and concept-wise thresholded experiments – TF-IDF weights outperform the raw frequency vectors. This probably suggests that TF-IDF is important for semantic frequency-based features, but only at very extreme dimensionality reductions. These findings show that the semantic vector-space features like the ones employed by our approach can be rather noisy, despite representing high-level information. Even though frequency-based concept-wise thresholding works well in our evaluation, both truncation schemes warrant further investigation (e.g., in the context of hypernymy activation propagation and context embedding configurations). It is also interesting to explore the effect of more sophisticated methods for dimensionality reduction on the enriched data produced by our pipeline.
- **Replacement versus concatenation:** The experimental results over both datasets indicate a considerable performance difference between replacing the word embedding with the semantic vector and concatenating the two. The latter consistently outperforms the former, demonstrating that WordNet-based concept information is less important than the distributional and statistical properties that are captured by the word embeddings. This finding carries over to macro-averaged F1-score results for the 20-Newsgroups dataset, but does not apply to the Reuters dataset, where replacing the word embedding yields the best macro-averaged score. This is an interesting divergence from the behaviour in 20-Newsgroups that is possibly explained by the very large class imbalance of Reuters. This means that the semantic concept-based features probably provide useful information for under-represented cases that are inadequately covered by the embedding training. In other words, the concept-based information, which can be readily extracted from a semantic resource, is more suitable for representing minorities (*i.e.*, under-sampled classes), counterbalancing

the lack of a critical mass of data that is required by embeddings. This hypothesis is reinforced by the comparison of our approach with vectors pre-trained on very large corpora, as discussed below (Section 5.1.8).

- **Using raw concept frequencies versus TF-IDF scores:** The experimental results over the 20-Newsgroups dataset contradict previous knowledge about the relationship of token-based weighting schemes, when applied on lexical terms (e.g. words / sentences / documents). There, discounting common terms consistently boosts the performance on the task at hand. Yet, the inverse relationship manifests itself in the world of concepts, with TF-IDF weights always underperforming the raw frequency vectors in terms of accuracy. It would be interesting to examine information gain measures on such concept-based features as an alternative method towards identifying the most suitable concepts for augmenting classification performance.

5.1.8.2 Comparison to the state-of-the-art

We now compare our approach against the state-of-the-art in the field, including both fitted and pre-trained embedding approaches; in all cases, the resulting vectors are used in the classification pipeline in the same way as all other configurations examined and described in this work. Table 5.12 reports the performance of the majority classifier, the “embedding-only” baseline method which uses only the fitted word2vec embedding, (i) the FastText [296] embeddings, which are fitted to each dataset (ii) the publicly available pre-trained versions of word2vec, FastText and GloVe [493] embeddings, with a dimensionality of 300, 50 and 50, respectively; (iii) the 50-dimensional single-sense vectors used in [262] for multi-sense vector generation; (iv) the main sense embedding approaches, *i.e.*, the cluster-based, multi-sense 50-dimensional word vectors from [262], the 400-dimensional SensEmbed vectors [270] and supersense embeddings [181]. For (iv), we use the publicly available pre-trained vectors of each method, while for (i), (ii) and (iii) we also apply retrofitting [174] using the same WordNet relations as in our semantic augmentation process. This process runs as a post-processing step for 10 iterations - we experimented with more iterations (up to 50), but observed no improvement.

We can see that our approach outperforms all other state-of-the-art word embeddings – not only the FastText and word2vec vectors that are fitted on the same dataset as our approach, but also the pre-trained versions of FastText, GloVe and word2vec embeddings, which have been trained on a far larger corpora. Only the 300-dimensional pre-trained word2vec surpasses the “embedding-only” baseline. Surprisingly enough, retrofitting the embeddings consistently results in inferior performance, both for the pre-trained ones and for those fitted from scratch. Regarding sense embeddings, both supersenses and SensEmbed vectors work well, surpassing the “embedding-only” baseline, but they do not outperform our approach. The multi-context cluster-based approach underperforms all other configurations.

Further, we include an additional evaluation on two more datasets.

1. The BBC corpus [224] contains articles from the news domain, annotating them

config	Reuters		20-Newsgroups	
system	accuracy	ma-f1	accuracy	ma-f1
majority baseline	0.290	0.005	0.005	0.053
embedding-only	0.725	0.295	0.724	0.716
our approach	0.749	0.378	0.784	0.790
other trained embeddings				
FastText [296]	<u>0.732</u>	0.319	<u>0.751</u>	<u>0.743</u>
FastText + retrofitting	0.717	0.260	<u>0.748</u>	<u>0.740</u>
word2vec + retrofitting	0.709	0.248	0.717	0.710
pre-trained embeddings				
glove [493]	0.702	0.275	0.620	0.610
glove + retrofitting	0.684	0.235	0.587	0.575
FastText	<u>0.733</u>	<u>0.310</u>	<u>0.734</u>	<u>0.727</u>
FastText + retrofitting	0.705	0.239	0.706	0.695
word2vec (300-dim)	<u>0.737</u>	0.311	0.721	0.712
word2vec (300-dim) + retrofitting	0.689	0.239	0.476	0.465
single-context [262]	0.661	0.227	0.541	0.531
single-context + retrofitting	0.629	0.175	0.464	0.454
pre-trained sense embeddings				
multi-context [262]	0.570	0.121	0.430	0.412
SensEmbed [270]	<u>0.728</u>	<u>0.308</u>	0.722	0.714
Supersenses [181]	<u>0.729</u>	<u>0.313</u>	<u>0.733</u>	<u>0.725</u>

Table 5.12: Dataset-wise comparison with the state-of-the-art in terms of accuracy and macro-F1 score. Underlined values outperform the “embedding-only” (50-dimensional fitted word2vec) baseline, while **bold** values denote column-wise maxima.

system	bbc		ohsumed	
	accuracy	ma-f1	accuracy	ma-f1
majority	0.230	0.075	0.172	0.013
embedding-only	0.970	0.970	0.384	0.300
ours	0.976	0.976	0.435	0.373
other pre-trained embeddings				
word2vec	<u>0.973</u>	<u>0.973</u>	0.307	0.244
word2vec + retrofitting	0.880	0.878	0.313	0.250
SensEmbed	0.969	0.969	0.328	0.215
Supersenses	0.852	0.851	0.229	0.148

Table 5.13: Evaluation of representative configurations on additional datasets. Underlined values outperform the “embedding-only” baseline, while **bold** values denote column-wise maxima.

with 5 categories (business, politics, entertainment, tech and sports). It consists of 2,225 samples and is evaluated via 10-fold cross validation. Mean document size is approximately 220.1 words.

2. The Ohsumed dataset [340] contains medical texts, specifically 10,433 train and 12,733 test samples, with an approximate mean document size of 108.1 words.

These corpora extend our evaluation to datasets with few classes and small number of samples (as is the case with BBC) and to datasets from a radically different domain (*i.e.*, the medical content of Ohsumed). Table 5.13 presents the experimental results over the additional datasets for the two main baselines, our best performing configuration as well as top performers from Table 5.12.

We observe that our approach performs the best for both the additional datasets, with the difference being less noticeable on the BBC dataset. The lexical-only word2vec pre-trained embeddings outperform both sense-based approaches, out of which SensEmbed achieves the highest accuracy. Retrofitting word2vec vectors improves the classification results to a minor extent over the Ohsumed dataset.

Regarding experimental results that are reported in relevant studies in the literature, a WordNet-based enrichment approach in [158] achieves macro F1-scores of 0.719 and 0.717 for Reuters and 20-Newsgroups, respectively. However, it considers only the 10 most populous classes of the Reuters dataset, unlike our evaluation, which uses all available classes (despite the respective class imbalance and folksonomy-based noise explained in Section 5.1.7.1). In [462], the authors apply a frequency-based semantic enrichment approach to heavily reduced training and test portions of the Reuters and 20-Newsgroups datasets, achieving 0.744 and 0.557 precision, respectively (they do not specify the aggregation type). The corresponding scores for our best approach for the Reuters dataset is 0.748 / 0.42 micro and macro average respectively, and a 0.8 micro-macro precision value for 20-Newsgroups. In [93], the authors examine generative DNN topic models that are able to capture and model semantic information in the form of document meta-

data. Applying their approach to document classification, they achieve an optimal accuracy of 0.71 over 20-Newsgroups, which is considerably lower than our top performance (0.79). In [502], the authors employ a combination of CNN and LSTM networks with a semantic augmentation process via sense and supersense embeddings extracted from WordNet or Wikipedia via BabelNet. Their best configuration (using WordNet supersense trained embeddings) achieves an F1-score of 0.858 on a subset of 6 classes from 20-Newsgroups, 0.22 on OhsuMed and 0.934 on the BBC dataset; our corresponding scores are 0.790 (on the entire 20-Newsgroups labelset of 20 classes), 0.373 and 0.976. To our knowledge, our approach is directly outperformed only by [284], where the authors use a DBN architecture with softmax classification, achieving an accuracy of 0.8688 and 0.8263 over Reuters and 20-Newsgroups, respectively. Their method does not use semantic information, but employs a larger and deeper neural model than ours together with embedding fine-tuning. We also note that they use a heavily reduced version of the Reuters dataset that consists of just 10 classes.

5.1.8.3 Execution runtime requirements

Finally, we note that the investigation and design of the experimental setup focused on improving performance and examining enrichment potential in an intuitive and explainable way. As a result, our empirical evaluation illustrates a clear trade-off. Advantages include improved classification performance, enhanced explainability and relatively low requirements with respect to the size of training data. On the other hand, enriched workflows of our approach may require large execution times – this is mainly caused by the large dimensionality of the knowledge-based feature vector, which in turn depends on the knowledge resource and dataset qualitative attributes that determine the volume of conceptual information extracted. Large feature vectors impose a considerable overhead in data processing operations, increase the required number of parameters of the neural classifier and may require longer training and prediction times. For instance, our enriched workflows in our investigation, while outperforming the embedding-only baselines, the total train and test pipeline was $7\times$ (BBC dataset) or $20\times$ (OhsuMed dataset) slower. Further research is required to optimize our proposed approach with respect to runtime performance. In this study, we examined naive frequency-based filtering methods for the knowledge-based features that, as already stated earlier in this section, produced very promising results. This indicates avenues for investigation towards improving the execution times of our approach - namely, via the looking into more sophisticated dimensionality reduction methods of enriched information.

5.1.9 Conclusions

This study focused on the semantic augmentation methods for text classification. To boost classification performance, we extract frequency-based semantic information from the WordNet semantic graph and fuse it with deep neural embeddings. This enrichment approach combines expressive content-based features with knowledge-based features to

create a powerful model that achieves both state of the art performance and improved, built-in explainability. Specifically, the experimental evaluation demonstrates the following findings:

1. The use of semantic information from resources such as WordNet significantly improves classification performance, when applied to the input space of the neural model. This straightforward approach outperforms more complex ones in the literature that use the semantic graph to augment the training of embeddings, or to post-process the embedding vectors via retrofitting.
2. Concatenating the embedding and semantic vectors works best, on average, with maximal gains coming at the cost of an increased dimensionality. This increases the computational complexity of learning the classification model. However, this can be radically alleviated by filtering the component concepts. This is best realized by a threshold on minimum concept frequency.
3. Using raw, unnormalized concept frequency scores works best, while the *basic* disambiguation strategy is sufficient to achieve the best performance (with POS performing comparably). This seems to further simplify the semantic augmentation process, but a deeper investigation is required on additional and diverse datasets.
4. Hypernymy n -th order propagation via a spreading activation mechanism reinforces the already superior semantic augmentation process to a statistically significant extent.
5. Our proposed approach generates errors that are understandable, explainable and intuitive, as indicated by an analysis of misclassification results.

Directions for possible further investigation for this work include examining issues raised from the findings of our experimental analysis, such as the behavior of TF-IDF weighting in frequency-based semantic vectors. Additionally, it would be useful to investigate the effect of dimensionality reduction approaches for this kind of vector space data, ranging from pooling aggregations (e.g. averaging, multiplication, etc.) to transformation methods such as PCA [292], LDA [186] or a neural embedding process, similar to other works [364]. The *context-embedding* approach could be tested on additional semantic resources, especially ones that provide a larger supply of example sentences per concept. Finally, semantically-augmented classification in conjunction with sequence-based neural classifiers could be explored, such as Long Short-Term Memory [252] and Gated Recurrent Unit [114] models, as well in cases where embeddings of a larger dimensionality are employed.

5.2 Enriching Embeddings for Automatic Text Summarization

5.2.1 Introduction and Overview

In recent years, the abundance of textual information resulting from the proliferation of the Internet, online journalism and personal blogging platforms has led to the need for automatic summarization tools. These solutions can aid users to navigate the saturated information marketplace efficiently via the production of digestible summaries that retain the core content of the original text [690]. At the same time, advancements introduced by deep learning techniques have provided efficient representation methods for text, mainly via the development of dense, low-dimensional vector representations for words and sentences [348]. Additionally, semantic information sources have been compiled by humans in a structured manner and are available for use towards aiding a variety of natural language processing applications. As a result, semantic augmentation approaches can introduce existing knowledge to the neural pipeline, circumventing the need for the neural model to learn all useful information from scratch.

In this study, we examine the effect of semantic augmentation and post-processing techniques on extractive summarization performance. Specifically, we modify the input features of a deep neural classification model by injecting semantic features, simultaneously employing feature transformation post-processing methods towards dimensionality reduction and discrimination optimization. We aim to address the following research questions.

- Can the introduction of semantic information in the network input improve extractive summarization performance?
- Does the semantic augmentation process benefit via dimensionality reduction post-processing methods?

The rest of this section is structured as follows. In section 5.2.2 we cover existing related work relevant to this study. This is followed by a description of our approach (section 5.2.3). In section 5.2.4 we outline our experimental methodology and discuss results and findings. Finally, we present our conclusions in section 5.2.5.

5.2.2 Related work

5.2.2.1 Text representations

Extensive research has investigated methods of representing text for Natural Language Processing and Machine Learning tasks.

Vector Space Model (VSM) approaches project the input to a n -dimensional vector representation, exploiting properties of vector spaces and linear algebra techniques for cross-document operations. Approaches like the Bag-of-Words [545] have become popular baselines, mapping the occurrence of an input term (e.g. a word) to their occurrence frequencies in the text. Modifications to the model include refinements in the term weighting strategy such as DF and TF-IDF normalizations [684, 544], term preprocessing such as

stemming and lemmatization [289], and others. Further, sentence and phrase-level terms are examined [563], along with n-gram approaches, which consider n-tuple occurrences of terms instead [82, 304, 510].

Other approaches encode term co-occurrence information via representation learning, relying on the distributional hypothesis [235] to capture semantic content. At the same time, the need to circumvent the curse of dimensionality [237] of term-weight feature vectors has led to the production of dense, rather than sparse representations. Early such examples used analytic matrix decompositions on co-occurrence statistics [292, 139, 257], while more recently, vector embeddings are iteratively optimized learned by analyzing large text corpora using local word context in a sliding window fashion [437, 440], or using pre-computed pairwise word co-occurrences [493]. More refined methods break down words to subword units [67], where learning representations for the latter enables some success in handling out-of-vocabulary words.

5.2.2.2 Extractive summarization

In summarization, contrary to the abstractive approach where output summaries are generated from scratch [690], the extractive method relies on sentence salience detection to retain a minimal subset of the most informative sentences in the original text [232]. VSM approaches have been widely utilized in sentence modelling for this task, with a variety of methods for determining term weights based on word frequency, probability, mutual information or TF-IDF features, sentence similarity, as well as a variety of feature combination methods [448, 423, 460, 188, 389]. Other popular handcrafted features used are syntactic / grammar information such as part-of-speech tags, as well as sentence-wise features such as sentence position and length. Finally, similarity scores to title, centroid clusters and predefined keywords can be used to score / rank sentences towards salience identification and extraction [461, 690].

Other works adopt a topic-based approach, using topic modelling techniques towards sentence salience detection. For example, the work in [23] builds topics via a clustering process, using a word and sentence-level vector space model and the cosine similarity measure. Clustering techniques have been applied to this end, for sentence grouping and subsequent salience identification [513].

Graph methods have also been exploited; In [345], the authors adopt a graph-based probabilistic language model towards building a topic hierarchy for predicting representative vocabulary terms. The MUSE system [376] combines graph-modelling with genetic algorithms towards sentence modelling and subsequent ranking, while the work in [434] builds sentence graphs using a variety of feature bags and similarity measures and proceeds to extract central sentences via multiple iterations of the TextRank algorithm.

5.2.2.3 Semantic enrichment

Semantic information has been broadly exploited towards aiding NLP tasks, using resources such as Wordnet [441], Freebase [68], Framenet [33] and others. Such external knowledge bases have seen widespread use, ranging from early works on expansion of rule-based discrimination techniques [562], to synonym-based feature extraction [529] and large-scale feature generation from WordNet synset relationships edges for SVM classification [413].

In extractive summarization, semantic information has been used as a refinement step in the sentence salience detection pipeline. For example, in [130], the authors utilize WordNet synsets as a keyphrase ranking mechanism, based on candidate synset relevance to the text. Other approaches [640] use semantic features from Wordnet and named entity extraction, followed by a PCA-based post-processing step for dimensionality reduction. Wordnet is also utilized in [359] where the authors use the resource for sentence similarity extraction, using synset similarity on the word level and treating the resulting scores as additional features for summarization and citation linkage.

Our approach bears some similarities with the work of [640], extending the investigation to additional post-processing techniques to PCA, examining post-processing application strategies, and adopting deep neural word embeddings as the lexical representation, while grounding on a number of baselines. In the following section, we will describe our approach in detail, including text representation, semantic feature extraction, training and evaluation.

5.2.3 Proposed Method

5.2.3.1 Problem definition

We formulate the task of automatic summarization as a classification problem. Given a document consisting of N sentences $D = \{s_1, s_2, \dots, s_N\}$ and a ground truth (extractive) summary of size k , $G = \{g_1, g_2, \dots, g_k\}$, $g_i \in D$, a classification-based extractive summarization system F selects salient sentences $P = \{p_1, p_2, \dots, p_k\}$ via $F(D) = P$, such that P is as close to G as possible. In this work, $F(\cdot)$ is a data-driven machine learning model, built by exploiting statistical features in the input text.

5.2.3.2 Text representation

We use a variety of approaches for representing the textual component of a sentence. First, we employ the Continuous Bag-of-Words (CBOW) variant of the popular word2vec model [440], which builds vector representations of a word using a statistical language model that predicts the word based on its surrounding context. More formally, given a center word in a sentence, w_c and a set of $2k$ context words around it $w_{context} =$

$[w_{c-k} \dots, w_{c-1}, w_{c+1}, \dots, w_{c+k}]$, CBOW tries to optimize the conditional probabilistic neural language model $P(w_c|w_{context})$.

We train embeddings from scratch with this method, optimizing with the cross-entropy loss, ending up with a vector representation for each word in the dataset. We subsequently produce the final, sentence-level representation by averaging the vectors corresponding to words in a sentence. In addition to embedding training, we examine the performance of pre-trained FastText [296] embeddings, produced by a model that captures subword information via character embeddings, enabling handling of out-of-vocabulary words. Additionally, we employ direct sentence-level modelling alternatives via the doc2vec [347] extension of word2vec, as well as a sentence-level TF-IDF baseline.

5.2.3.3 Semantic representation

In order to capture and utilize semantic information in the text, we use the WordNet semantic graph [441], a lexical database for English, often used as an external information source for machine learning research in classification, summarization, clustering and other tasks [268, 158, 383, 449, 44, 130, 478]. In Wordnet, semantic relations between concepts are captured in a semantic graph of synonymous sets (*synsets*), as well as multiple types of relations of lexical / semantic nature, such as hypernymy and hyponymy (is-a graph edges), meronymy (part-of relations, and others). We employ WordNet as an enrichment mechanism, extracting frequency-based features from corpus words. Specifically, we mine semantic concepts from each word in the text, arriving at a sparse high-dimensional bag-of-concepts for each document. This vector is concatenated to the lexical representation. To deal with the curse of dimensionality [237] of this approach, we apply dimensionality reduction via PCA [292], LSA [139] or K-Means [391]. We apply each transformation on two settings; first, the semantic information channel is reduced, then concatenated with the sentence embedding vector. Alternatively, we apply the reduction on the concatenated, enriched vector itself.

5.2.4 Experiments

5.2.4.1 Datasets

We use the english version of the Multiling 2015 single-document summarization dataset [207, 122]¹⁷ for our experimental evaluation. The dataset is built from wikipedia content, consisting of articles paired with a number of human-authored summaries. For each of 40 languages, 30 documents and summary sets are provided.

In this work, we focus on the English version of the dataset, due to our reliance on word embedding features, which are predominantly available for the English language. In addition, we apply two preprocessing steps. First, we reformat the ground truth towards an

¹⁷<http://multiling.iit.demokritos.gr/pages/view/1516/multiling-2015>

feature	train	test
document sentences	233	184.9
document summary sentences	77.9	13.5
document words	25.5	22.8
samples	6990	5546

Table 5.14: Details of the Multiling 2015 single-document summarization dataset. All values are averages across documents, except for the number of samples.

extractive summarization setting, since the provided summaries are written from scratch. Specifically, we annotate source sentences with an extractive summarization binary label $l \in \{0, 1\}$ (e.g. 1 if it is a member of the extractive summary and 0 otherwise). This is accomplished via the following steps. First, for each provided summary sentence p_i , we rank source sentences $s \in S$ with respect to the n-gram overlap with p_i , after stopword filtering and excluding already positively-labelled sentences $s_j \in S : l_j = 1, i \neq j$. The top-ranked source sentence is matched to the ground truth summary sentence, and considered to be a member of the extractive summary. Secondly, in an effort to address the severe imbalance that results from the modifications of previous step (i.e. class 0 being 13 to 14 times more populous than class 1), we oversample positively labelled sentences for each document, arriving at a 2 : 1 negative to positive ratio, at most.

After these steps, we end up with the final version of the dataset which is described in detail in table 5.14. Having a sentence-level label for summary meronymy, we can thus produce the final summary by concatenating the positively classified sentences. It should be noted that via this setting, evaluating candidate summaries with the dataset provided ground truth summaries implies a minimum performance penalty. This is reported in the results in the succeeding section 5.2.4.3.

5.2.4.2 Setup

We train embeddings with the word2vec CBOW variant using gensim [522]. We run the algorithm for 50 epochs, on a 10-word window, maintaining a minimum word frequency threshold of 2 occurrences in the training text. We produce 50-dimensional embeddings via this process. In addition, we use the publicly available¹⁸, 300-dimensional pre-trained FastText embeddings for the corresponding configuration.

To set up the deep neural classifier, we run a grid search on the number of layers (ranging from 1 to 5) and layer size (ranging from 64 to 2048) for a multilayer perceptron architecture, using a 5-fold validation scheme. This process illustrated a 5-layer architecture of 512-neuron layers as the best performing, and is the one we adopt for all subsequent experiments. This architecture is trained for 80 epochs, reducing the learning rate on an adaptive learning rate reduction policy and maintaining an early-stopping protocol of 25 epochs.

¹⁸<https://fasttext.cc/>

Using this learning framework, we test each candidate configuration using a 5-fold validated scheme, reporting mean measure values as the overall result. For all measures, the cross-fold variance stayed below $10e - 4$ and is omitted. The Keras machine learning library¹⁹ is used for building and training the neural models.

5.2.4.3 Results and discussion

Tables 5.15 and 5.16 present experimental results for the evaluation of semantic augmentation on word2vec and FastText embeddings, respectively. Each configuration is evaluated in terms of micro and macro F1 score (`mi-F` and `ma-F` columns, respectively), with respect to classification performance of the oversampled dataset (as detailed in 5.2.4.1). In addition, we measure Rouge-1 and Rouge-2 scores of the final composed summary (stitched together from positively classified input sentences) with respect to the handwritten ground truth summary provided in the dataset. Since the difference between the latter two guarantees a minimum error (see 5.2.4.1), we report the best possible performance in the `gt` configuration, depicting performance for each evaluation measure when sentence classification is perfect. In addition, via the `prob` configuration we report a probabilistic baseline classifier, which decides based on the label distribution in the training data. Moreover, token frequency-based baselines – namely bag-of-words (BOW) and TF-IDF [544] – are reported in the `BOW` and `TF-IDF` rows. Lexical-only and semantically-augmented baseline runs are reported as `x` and `x-sem` respectively, where $x \in [w2v, fasttext]$. Finally, the effect of each post-processing method on the semantically augmented baseline is illustrated, where a configuration of `+conf-N` denotes a vector post-processing method `conf` that produces N -dimensional vectors. The resulting vector dimension that is fed to the classifier is reported in the column `dim`, and the different semantic augmentation post-processing methods are denoted by `tc` – i.e. first transform the semantic channel, then concatenate to the embedding – and `ct` – i.e. concatenate the semantic vector to the lexical embedding, then apply the transformation.

Regarding word2vec trained embeddings (table 5.15), we can see that introducing semantic information improves macro F1, Rouge-1, Rouge-2 performance. Compared with the bag-based baselines, we observe the word2vec CBOW embeddings yielding worse micro F1 performance than both bag approaches, but considerably better Rouge scores. In addition, the semantically enriched `w2v` configuration outperforms the bag approaches in macro-F1 score and the examined Rouge measures.

In general, we observe that micro-F1 scores appear to be less reliable measures in this setting, given the considerable large class imbalance of the dataset. This is apparent in the baseline `w2v` and `w2v-sem` baseline runs, however the effect is most pronounced in k-means configurations for dimensions greater than 50, where the best micro-F1 score is encountered, but the performance of all other metrics is degenerate. This is understandable, since cases where the classifier completely relies on the majority class (0, or “non-summary sentences” in our case), it can converge to a state characterized by a total

¹⁹<https://keras.io/>

lack of positively classified sentences. This in turn produces zero rouge scores and sub-chance macro-averaged F1 scores, which is the case observed for these configurations. The best-performing configuration turns out to be LSA with 500-dimensional vectors, with regard to Rouge-1 and Rouge-2 scores, with the 100-dimensional PCA configuration performing best in terms of macro F1.

Regarding comparison between the two post-processing strategies, we can observe that t_c appears to be working slightly better when measuring micro-F1 scores, but in terms of macro-F1 and Rouge scores, concatenating prior to post-processing works considerably better. This is not surprising, as the transformation of the bimodal vector into a common, shared space can be expected to be a far more efficient fusion of the lexical and semantic channels, compared to simple concatenation.

Regarding FastText-based runs, a similar baseline performance is observed. Bag-based baselines achieve the best micro-F1 score, but inferior results in all other measures. Similarly to word2vec, the lexical-only FastText run achieves better F1 scores, however the semantically enriched embedding fares far better in terms of Rouge-1 and Rouge-2 performance. Likewise, similar behavior is observed with regard to post-processing and concatenation order and the usefulness of the micro-F1 score compared to the other measures. Notably, the 50-dimensional LSA performs well with the t_c strategy, while an analogous degenerate behaviour is evident with the K-means configurations. As in the word2vec run, the 500-dimensional LSA produces the best macro-F1 and Rouge scores.

Comparing the word2vec and FastText-based runs, we can observe the word2vec configurations (trained on the target dataset from scratch) achieve better Rouge-1 and Rouge-2 scores than the pre-trained FastText embeddings, on the best configurations of both baseline and best performing post-processed configurations (500-dimensional LSA).

In light of these results, we return to the research questions stated in the beginning of this document.

- **Can the introduction of semantic information in the network input improve extractive summarization performance?**

It appears that the introduction of semantic information can introduce benefits to the extractive summarization pipeline. This is illustrated by the Rouge scores, which are considerably improved in the augmented configurations, for both embeddings examined. On the contrary, micro / macro-F1 results are either not significantly affected or can even deteriorate. However, as discussed above, we argue that the severe class imbalance of the dataset makes these measures less indicative of system performance, compared to Rouge.

- **Does the semantic augmentation process benefit via dimensionality reduction post-processing methods?**

The augmentation process can improve with post-processing methods. This is expected, since the sparse bag-based semantic vectors are bound to contain noise and/or redundant and overlapping information that will affect the learning model further down the summarization pipeline. For both embeddings examined, such con-

config	dim	mi-F		ma-F		Rouge-1		Rouge-2	
gt	N/A	1.000		1.000		0.414		0.132	
prob	N/A	0.871		0.501		0.051		0.009	
BOW	15852	0.9254		0.5131		0.094		0.017	
TF-IDF	15852	<u>0.9260</u>		0.5122		0.085		0.015	
w2v	50	0.923		0.508		0.151		0.027	
w2v-sem	10292	0.906		<u>0.519</u>		<u>0.260</u>		<u>0.048</u>	
config	dim	tc	ct	tc	ct	tc	ct	tc	ct
+lsa-50	100	<u>0.9225</u>	0.9214	<u>0.5223</u>	0.5222	0.166	<u>0.195</u>	0.030	<u>0.036</u>
+lsa-100	150	<u>0.9202</u>	<u>0.9207</u>	0.5164	<u>0.5217</u>	0.188	<u>0.202</u>	0.038	<u>0.038</u>
+lsa-250	300	<u>0.9197</u>	0.9165	0.5198	<u>0.5289</u>	0.181	<u>0.246</u>	0.037	<u>0.040</u>
+lsa-500	550	<u>0.9218</u>	0.9053	0.5190	<u>0.5337</u>	0.159	0.305	0.030	0.059
+pca-50	100	<u>0.9208</u>	0.9101	0.5195	<u>0.5329</u>	0.193	<u>0.242</u>	0.039	<u>0.049</u>
+pca-100	150	<u>0.9207</u>	0.9141	0.5206	0.5349	0.178	<u>0.234</u>	0.036	<u>0.047</u>
+pca-250	300	<u>0.9217</u>	0.9146	0.5217	<u>0.5250</u>	0.171	<u>0.237</u>	0.035	<u>0.044</u>
+pca-500	550	<u>0.9223</u>	0.9107	0.5202	<u>0.5254</u>	0.161	<u>0.255</u>	0.032	<u>0.049</u>
+kmeans-50	100	0.9089	<u>0.9257</u>	0.5267	0.4821	<u>0.252</u>	0.018	0.056	0.005
+kmeans-100	150	0.9028	0.9272	<u>0.5107</u>	0.4811	<u>0.133</u>	0.000	0.028	0.000
+kmeans-250	300	0.9272	0.9272	0.4811	0.4811	0.000	0.000	0.000	0.000
+kmeans-500	550	0.9272	0.9272	0.4811	0.4811	0.000	0.000	0.000	0.000

Table 5.15: Experimental results on the MultiLing 2015 MSS dataset using 50-dimensional word2vec embeddings. Bold values indicate maxima across rows for that column. Underlined values correspond to an improvement over the counterpart configuration (tc versus ct, or x versus x-sem).

config	dim	mi-F		ma-F		Rouge-1		Rouge-2	
gt	N/A	1.000		1.000		0.414		0.132	
prob	N/A	0.871		0.501		0.051		0.009	
BOW	15852	0.9254		0.5131		0.094		0.017	
TF-IDF	15852	0.9260		0.5122		0.085		0.015	
fasttext	300	0.923		<u>0.517</u>		0.156		0.029	
fasttext-sem	10542	0.919		0.516		<u>0.204</u>		0.043	
config	dim	tc	ct	tc	ct	tc	ct	tc	ct
+lsa-50	350	0.9167	<u>0.9214</u>	<u>0.5231</u>	0.5195	<u>0.206</u>	0.182	<u>0.038</u>	0.032
+lsa-100	400	0.9200	<u>0.9212</u>	0.5196	<u>0.5224</u>	0.171	<u>0.189</u>	0.032	0.036
+lsa-250	550	<u>0.9237</u>	0.9134	0.5221	<u>0.5370</u>	0.145	<u>0.278</u>	0.031	0.053
+lsa-500	800	<u>0.9243</u>	0.9083	0.5201	0.5373	0.128	0.296	0.025	0.056
+pca-50	350	<u>0.9186</u>	0.9145	0.5205	<u>0.5319</u>	0.182	<u>0.234</u>	0.036	0.045
+pca-100	400	<u>0.9208</u>	0.9160	0.5187	<u>0.5369</u>	0.160	<u>0.230</u>	0.037	0.044
+pca-250	550	<u>0.9233</u>	0.9146	0.5210	<u>0.5286</u>	0.189	<u>0.229</u>	0.038	0.045
+pca-500	800	<u>0.9239</u>	0.9096	0.5223	<u>0.5261</u>	0.152	<u>0.255</u>	0.032	0.047
+kmeans-50	350	0.8995	0.9238	<u>0.4928</u>	0.4833	<u>0.071</u>	0.022	<u>0.018</u>	0.006
+kmeans-100	400	0.8903	0.9272	<u>0.4897</u>	0.4811	<u>0.071</u>	0.000	<u>0.018</u>	0.000
+kmeans-250	550	0.9272	0.9272	0.4811	0.4811	0.000	0.000	0.000	0.000
+kmeans-500	800	0.9272	0.9272	0.4811	0.4811	0.000	0.000	0.000	0.000

Table 5.16: Experimental results on the MultiLing 2015 MSS dataset using 300-dimensional FastText embeddings. Bold values indicate maxima across rows for that column. Underlined values correspond to an improvement over the counterpart configuration (tc versus ct, or x versus x-sem).

figurations improve upon the baseline and achieve the best scores, for all evaluation measures included.

LSA-based transformations achieve top Rouge performance for both embeddings covered, as well as top F1 scores for the FastText embedding, with its frequency-based decomposition appearing to work better than PCA analysis. On the contrary, K-means clustering mostly failed to capture underlying semantic content into meaningful groups, especially for higher dimensions examined. Additionally, the post-processing transformation methods work best mostly when applied to the concatenated lexical and semantic vectors, rather than transforming the semantic information alone and then concatenating.

Apart from the specific research questions, it is notable that the large class imbalance has to be carefully handled, as – even with the dataset oversampling measures taken – the sentence classifier can converge into degenerate cases, as was the case with the higher dimensional configurations of K-means.

At this point, we note that since our system does not account for selected sentence order, we limit our comparison of each approach to only the gt configuration, rather than the human-authored summaries; even for cases with perfect classification performance, the results are far from optimal (e.g. Rouge 1, Rouge 2 scores of 1.0) since there is no guarantee that sentence order is preserved in the extractive ground truth generation, detailed in 5.2.4.1. This introduces an upper bound to performance and prevents meaningful comparison to related work. Instead, this study illustrates the contribution of semantic information to the pipeline, as illustrated above.

As a last note, we compare our results with respect to the unaltered, human-written summaries – i.e. which are not composed of input sentences as per the extractive setting, after reiterate that our preliminary approach does not take into account sentence order or target length. First, the gt extractive ground truth we generated achieves an Rouge-1 and Rouge-2 score of 0.245 and 0.57 respectively, effectively serving as an upper bound for our performance. The best-performing 500-dimensional LSA configuration for word2vec trained embeddings performs at 0.196 and 0.015 for Rouge-1 and Rouge-2, respectively, and 0.191, 0.014 for FastText. These results fall short of the system performance levels on previous MultiLing community tasks [207], however the goal of this investigation was solely to illustrate the utility of the semantic component; future work (outlined below) plans on addressing this issue and align our results toward related work comparability.

5.2.5 Conclusions

In this work, we investigated the contribution of semantically enriching word embedding-based approaches to extractive summarization. Pre-trained embeddings as well as embeddings trained from scratch on the target dataset were utilized. For the semantic channel, frequency-based concept information from Wordnet is extracted, post-processed with a range of feature transformation and clustering methods prior or after concatenation with the lexical embeddings. A wide evaluation was performed on multiple configuration com-

binations and transformation dimensions, using micro/macro F1 and Rouge-1/Rouge-2 scores. Initial results show such semantic augmentation approaches can introduce considerable benefits to baseline approaches in terms of macro F1, Rouge-1 and Rouge-2 scores, with micro-F1 deemed inadequate for highly imbalanced problems such as the extractive summarization setting examined here. LSA-based decomposition works best out of the variants examined, outperforming PCA and K-means post-processing in terms of Rouge. In the future, more sophisticated transformation methods could be explored, such as encoder-decoder schemes via recurrent neural networks [252], dynamically fusing word embeddings into a sentence encoding and eliminating the need for word averaging in sentence-level vector generation. Alternatively, sequence-based classification could be explored in a similar fashion. Moreover, higher transformation dimensions could be covered, given the best configuration examined lies on the highest end of the examined range (500) and additional semantic resources can be utilized, via the bag-based approach used in this study, or by alternative methods of semantic vector generation [174]. Finally, the natural next step in this study would be the application of our semantic augmentation approach with a sentence ranking and a target length constraint mechanism, in order to make the results of the pipeline fairly comparable to related summarization systems.

5.3 Conclusions and Findings

This chapter presented variants of our proposed approach for representation enrichment. The approach implements deep representation learning with the input modification approach, as described in the content-based and enrichment chapters 2 and 4, respectively. It combines deep neural content-based features with encoded human knowledge from Wordnet, considering multiple techniques for disambiguation, embedding type and information spread. In summary, we emphasize the following findings and observations:

- The conducted large-scale experimental evaluation showcases performance improvements over baseline content-based configurations. This shows that knowledge infusion is beneficial to machine learning performance, in the cases of the classification and automatic summarization tasks.
- The proposed enriched representation method for classification achieves state of the art results with respect to competing modern systems. This indicates that utilizing structured human knowledge for augmenting representations is a promising avenue towards enhancing performance and robustness of classification systems.
- The modularity of the architecture in the proposed approach presents multiple points of improvement, avenues for extensions and modification, where future research work will have the potential to push performance and utilization of available knowledge even further.
- Our approach utilizes resources of human knowledge and knowledge-based features in a way that enhances the explainability and understanding of systems, facilitated by the use of intuitive, documented semantic features.

In the next chapter, we consolidate and exploit the research findings of this thesis into the implementation of a novel, real-world machine learning application.

6. APPLICATION IN THE INDUSTRY

In this chapter, we realize the body of conducted research into a Hate Speech Detection (HSD) system over social media content, applied and deployed in a real-world, industrial setting. We provide a coarse description of the methodology, as well as exploratory directions taken to arrive at a best-performing model configuration.

6.1 Problem Definition and Goals

The task of interest is a multiclass classification problem, approached from a machine learning perspective: given a dataset D consisting of N tuples (t_i, l_i) , $i = [1 \dots N]$, where t_i are short texts from social media and l_i are annotations corresponding to hate speech categories. We seek a classifier $F(\cdot)$ to learn to assign short texts with hate speech tags, i.e. $F(t_i) = \hat{l}_i$, with $l_i = \hat{l}_i$ for $i = [1, \dots, N]$, in the optimal case. In order to achieve robust performance for applicability in real-world settings, the adopted solution should:

- Take advantage and consider rich data within the scope of the task, to be able to achieve good generalization ability.
- Be easily extensible and configurable, to be able to adapt to changing and evolving needs in the industry
- Rigorously fine-tuned to optimal hyperparameters to achieve the best possible performance. Additionally, the fine-tuning process should be easy to monitor and track, as well as be interpretable by non-technical personnel to aid decision-making.

Given the aforementioned task and solution requirements, we move on to describe the dataset resources gathered to build the HSD system (section 6.2).

6.2 Dataset

The task of interest detects types of hate speech content. The types of interest refer to content that expresses stereotypes, generalizations and/or chauvinistic views that are examples of:

- Racism, i.e. related to the race of an individual or a group
- Sexism, related to expressions of misogyny / misandry
- Sexual orientation, connected to sexual orientation and gender identity
- Religious chauvinism, referring to the religion of an individual or a group

label	train		test	
	# instances	mean # words	# instances	mean # words
racism	2448	14.22	15	14.0
sexism	4213	15.57	15	17.53
orientation	677	12.78	15	12.47
religion	581	19.18	15	20.0
none	7761	13.99	15	16.53
overall	15680	14.59	75	16.11

Table 6.1: Development dataset for the HSD classifier

We additionally include a class for non-hateful content, to aid discrimination of instances that convey hate, rather than neutral or merely offensive texts.

To build the dataset, we use publicly available datasets annotated with labels of Hate Speech and consisting of texts predominantly from Twitter. Additionally, we obtain data with crawling techniques, e.g. via the Twitter python API ¹, using lists of relevant post IDs. We subsequently merge partial results from all the different sources via a conversion and curation procedure that applied data cleaning, label mapping and relevant information extraction. After this process, we arrive at the 5 labels of interest and a dataset of 15680 instances. Details of the corpus are illustrated on table 6.1.

6.3 Proposed Method

Given the aforementioned task and solution requirements, we proposed a system with the following components:

- The lexical component, which processes lexical content-based information of the input.
- The semantic component, which utilizes high-level semantic and lexicon-based knowledge.
- The learner component, which uses lexical and semantic information to fit the learning model.

Additionally, we use modern tools to aid model utilization, monitoring and fine-tuning, covered in section 6.4.

6.3.1 Lexical Processing

The lexical processing component of the HSD system deals with text preprocessing and lexical representation extraction from input instances of social media texts. For prepro-

¹<https://pypi.org/project/twitter/>

cessing, applied sentence and word tokenization to break apart social media texts into word tokens. For vocabulary reduction, we investigated the application of established preprocessing rules, such as replacing platform-specific artifacts (e.g. mentions, hashtags) and urls with static tokens.

Regarding text representations, we evaluated the following approaches:

- Word Embeddings (WE) – we used the embedding approaches of Word2Vec [440], GloVe [493] and FastText[296], in order to map word tokens in an input instance to real valued vectors $v \in R^d$, using average aggregation to represent the contents of the entire text as a d -dimensional vector. We experimented with publicly available pre-trained vectors²³⁴, as well as training the vectors from scratch on the training portion of the development dataset.
- Bag of Words (BOW) – the BOW approach maps input texts into a $|V|$ -dimensional vector, where V is the vocabulary that comprises the training portion of the development dataset. To deal with the large vector dimensionality, we experimented with retaining limited portions of the vocabulary as a computational cost / detection performance trade-off.

6.3.2 Semantic Enrichment

For semantic enrichment we evaluated both domain-specific and linguistic, domain-agnostic information. Each lexical representation was augmented with the following types of information:

- Conceptual information: we use version 3.0 Wordnet [441] as a source of semantic information: given a word w in an input text, we retrieve a synset from the ontology, defaulting to the first item in the acquired collection. This corresponds to the synset that represents the most frequent sense of the input word w .
- Domain-specific information: we compiled a large, manually curated collection of K keywords that convey hateful content, designed to correspond to the labelset of interest described in section 6.2. We expect that activations in these semantic features will contribute considerably to the richness of information passed down the HSD pipeline.

We utilize these sources of conceptual information to build bag-of-semantics vectors, subsequently concatenated to the vectorized output of the lexical component; this results in an semantically enriched representation, fed to the learning component (described in the next section, 6.3.3) to facilitate classification.

²<https://code.google.com/archive/p/word2vec/>

³<https://nlp.stanford.edu/projects/glove/>

⁴<https://fasttext.cc/docs/en/english-vectors.html>

6.3.3 Learning approach

For the learning model we investigate multiple approaches; for each, we train with L2 regularization over a max number of epochs, E :

- Logistic Regression (LR) [665] is a linear method for binary classification. We use the 'one-vs-rest' technique [17] to facilitate multiclass classification.
- Neural Networks (NN) [221], where we focus on feed-forward, fully-connected architectures. investigating configurations of N_l layers, each with N_n neurons. The model is trained with a batch size of 200 instances, with a learning rate of 0.001 and Adam optimization [312].

6.4 Tuning and Monitoring

We perform large-scale hyper-parameter tuning to arrive at a suitable model variant, in terms of computational efficiency and classification performance. We use Ray tune [369], a suite for scalable hyperparameter tuning, monitoring and tracking, optimized for execution in distributed environments, to tune our model. Some examined hyper-parameters considered include:

- Preprocessing techniques (e.g. extent of tweet token preprocessing)
- The lexical representation model type (e.g. WEs or BoW) and its parameters (e.g. vocabulary size in BoW, WE model (Word2Vec or FastText) pre-trained or from-scratch WEs, and WE vector dimensionality)
- The learning model (e.g. LR or NNs) and architecture / training parameters (e.g. total number of epochs E , number of NN layers and/or neurons)

Additionally, we experiment with different over/under-sampling configurations. We use MLflow [698], a tool for development monitoring, performance and version tracking and deployment, to organize training and tuning results. The tuned HSD model is served with Flask [226], a library for robust RESTful endpoint development and deployment.

6.5 Implementation, Development and Deployment

A modular codebase in python 3.8 was built to support the HSD task, including highly robust and extensible model development, configurable tuning and easy deployment and testing. We used scikit-learn [489] for learning, nltk [394] for text preprocessing and semantic augmentation, and tweet-preprocessor⁵ for Twitter-focused text preprocessing.

⁵<https://pypi.org/project/tweet-preprocessor/>

Development progressed in multiple phases, receiving feedback from the participating company and implementing requested use cases and relevant refactors, delivering a code repository of the final version upon completion.

6.6 Contributions

To summarize, our contributions for the industrial component of the study consists of the utilization of findings from the research component of the thesis into a Hate Speech Detection system. State of the art methods for representation enrichment were employed to build a learning model for the HSD task. A large-scale optimization and hyper-parameter tuning ensured we produced a fitting instantiation of the model, to be delivered for integration with products and workflows in the industry. We took great care in utilizing modern tools and approaches in Machine Learning pipeline development and following software engineering methodologies that build extensible and easy to use solutions for real world applications.

7. CONCLUSIONS AND FUTURE WORK

In this chapter we present our conclusion of the study conducted in this thesis. We provide a summary of the research work and findings in 7.1 and present directions for future work in section 7.2.

7.1 Summary and Contributions

In this thesis, we investigated the augmentation of Machine Learning systems with external resources of existing information. We retained a focus on the data representation component of the learning pipeline, adopting a broad view over learning models, data modalities and learning problems.

Our study began in chapter 2, where we examined content-based representation techniques for the classification problem, covering different data modalities (text, images and audio) and grouping methods with respect to sophistication of generated representations. Along with the literature survey, we provided a comparison between each representation extraction strategy, identifying pros and cons and a trend towards semantically rich features. We continued in chapter 3, where we investigated performance and robustness of proposed modifications, applications and/or extensions of multiple representation approaches for different tasks of interest, namely Hate Speech Detection, Automatic Document Summarization, Clustering / Event Detection and Multimodal Video Classification. The conducted research and literature review highlighted the need for improving representations as a means of enhancing task performance; an effective avenue for such improvements was argued to be utilizing resources of existing knowledge in chapter 4. There, we presented potential areas of impact knowledge-based enrichment can affect and provided a detailed literature review of different types of enrichment approaches along with the resources exploited in each such study (ontologies, knowledge bases, lexicons, etc.). Research insights and identified limitations led to the implementation and evaluation of proposed knowledge utilization methods, based on the enrichment of features extracted from deep representation learning algorithms with knowledge from the Wordnet resource. We argued in favor of performance and explainability gains of such an approach, which was applied in text classification and automatic summarization, achieving state-of-the art results and generating useful findings, all presented in chapter 5. Finally, a realization of the conducted research into a novel Hate Speech Detection system was outlined in chapter 6.

In light of the presented body of work, knowledge utilization in classification tasks shows promise; the proposed approaches, along with ongoing research efforts in the scientific community continue to invent novel enrichment techniques and improve existing approaches for knowledge injection across representation paradigms, with these efforts bearing fruit in classification performance gains. An example of the continued and persistent utilization of enrichment is its successful combination and integration with deep learn-

ing methods, as was showcased by the proposed methods in this thesis. Such avenues enable the exploitation of knowledge in representation learning in a holistic manner, and allows leveraging both content-based and high-quality structured information in state of the art classification pipelines. We believe that knowledge-aware deep learning approaches show promise in the search of optimally fusing distributional and knowledge-oriented information, exploiting the best of both worlds towards efficient and robust classification.

Currently, knowledge enrichment techniques heavily rely on resources of linguistic nature, since such resources can encapsulate high-quality and detailed semantic information. This has limited representation enrichment methods to mostly text data applications, with image and audio classification enrichment techniques having to use these knowledge resources indirectly – that is, via exploiting metadata, labelset and ontology-related information through their textual descriptions and lexicalizations. Based on our investigation in this study – specifically of end-to-end knowledge aware systems – the aforementioned limitation may have affected knowledge injection in image / audio classification pipelines to the (comparatively) simpler approaches of input and refinement-oriented techniques.

Further, in machine learning there exists a duality between learning and forming a representation itself: finding good representations for a given problem implies facilitating learning. This duality has been further accentuated through deep learning approaches [6, 85] and emphasized in this thesis. Given this relation, one would expect that the requirement for explainability that is apparent in the learning process could be also directly focused on the representations themselves. One may argue that enriching neural representations with external (explainable) knowledge may be able to affect the explainability of the enriched spaces. To this end, producing knowledge resources suitable for direct exploitation in image and audio representation enrichment would be a step towards better interpreting of large deep learning models in these domains.

7.2 Future Work

There are multiple ways to complement / extend the work in this thesis. Our primary focus was augmenting the representation component of a Machine Learning pipeline – one avenue for future work would be the investigation of enrichment approaches that focus on the learning algorithm, i.e. independent of the representation approach. Given aforementioned limitations in knowledge resources, we maintained a focus on the textual modality for the development of our proposed methods. Extensions to this could explore the utilization of input-modification enrichment to deep features of image and/or audio data. This could be achieved through auxiliary metadata that may accompany such instances (e.g. captions, tags, etc.) that can be used to mine Wordnet or other similar knowledge repositories, or by investigating an entirely different approach for matching knowledge to multimedia content. Additionally, an exploration of knowledge utilization for additional learning tasks (i.e. other than classification and automatic summarization) or even in a task-agnostic setting, would be beneficial towards arriving at a better understanding of the wider impact of knowledge-based enrichment. Finally, the proposed approach makes use

of well-defined knowledge-based features that improve the explainability of the data representation and learning components of the machine learning pipeline; the development of automated tools and utilities that exploit the definitions, documentation and grounding of these conceptual information units could further enhance the explainability of our semantically augmented learning framework.

ABBREVIATIONS - ACRONYMS

ML	Machine Learning
LR	Logistic Regression
SVM	Support Vector Machine
kSVM	Kernel Support Vector Machine
NB	Naive Bayes
NN	Neural Network
MLP	Multilayer Perceptron
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
LSTM	Long Short Term Memory
WE	Word Embeddings
BOW	Bag of Words
TF-IDF	Term Frequency - Inverse Document Frequency
VQ	Vector Quantization
PCA	Principal Component Analysis
QDA	Quadratic Discriminant Analysis
LSA	Latent Semantic Analysis
pLSA	Probabilistic Latent Semantic Analysis
LSI	Latent Semantic Indexing
SIFT	Scale-Invariant Feature Transform
SURF	Speeded-Up Robust Features
LBP	Local Binary Pattern
SPM	Spatial Pyramid Matching
NGG	N-Gram Graphs
GMM	Gaussian Mixture Modeling
VLAD	Vectors of Locally Aggregated Descriptors

MFCC	Mel Frequency Cepstral Coefficients
MC	Multiclass
SL	Single-Label
BIN	Binary
KNN	K Nearest Neighbors
HSD	Hate Speech Detection
DT	Decision Trees
GBDT	Gradient Boosted Decision Trees
LDA	Latent Dirichlet Allocation
RF	Random Forests
POOL	Pooling
NORM	Normalization

APPENDIX A. APPENDIX

A.1 20Newsgroups and Reuters dataset label names

In Table A.1, we list the mapping between label index and label name, for each dataset examined in the main experimental evaluation of the text classification enrichment study (section 5.1).

20-Newsgroups					
index	name	index	name	index	name
0	alt.atheism	7	rec.autos	14	sci.space
1	comp.graphics	8	rec.motorcycles	15	soc.religion.christian
2	comp.os.ms-windows.misc	9	rec.sport.baseball	16	talk.politics.guns
3	comp.sys.ibm.pc.hardware	10	rec.sport.hockey	17	talk.politics.mideast
4	comp.sys.mac.hardware	11	sci.crypt	18	talk.politics.misc
5	comp.windows.x	12	sci.electronics	19	talk.religion.misc
6	misc.forsale	13	sci.med		
Reuters					
index	name	index	name	index	name
0	acq	30	hog	60	platinum
1	alum	31	housing	61	potato
2	barley	32	income	62	propane
3	bop	33	instal-debt	63	rand
4	carcass	34	interest	64	rape-oil
5	castor-oil	35	ipi	65	rapeseed
6	cocoa	36	iron-steel	66	reserves
7	coconut	37	jet	67	retail
8	coconut-oil	38	jobs	68	rice
9	coffee	39	l-cattle	69	rubber
10	copper	40	lead	70	rye
11	copra-cake	41	lei	71	ship
12	corn	42	lin-oil	72	silver
13	cotton	43	livestock	73	sorghum
14	cotton-oil	44	lumber	74	soy-meal
15	cpi	45	meal-feed	75	soy-oil
16	cpu	46	money-fx	76	soybean
17	crude	47	money-supply	77	strategic-metal
18	dfl	48	naphtha	78	sugar
19	dlr	49	nat-gas	79	sun-meal
20	dmk	50	nickel	80	sun-oil
21	earn	51	nkr	81	sunseed
22	fuel	52	nzdlr	82	tea
23	gas	53	oat	83	tin
24	gnp	54	oilseed	84	trade
25	gold	55	orange	85	veg-oil
26	grain	56	palladium	86	wheat
27	groundnut	57	palm-oil	87	wpi
28	groundnut-oil	58	palmkernel	88	yen
29	heat	59	pet-chem	89	zinc

Table A.1: Label indexes to name mapping, for the per-label performance graphs for the 20-Newsgroups and Reuters datasets in the study in section 5.1.

BIBLIOGRAPHY

- [1] George A. Miller, R Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J. Miller. Introduction to WordNet: An On-line Lexical Database*. 3, 1991. 85
- [2] Martí Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, and Matthieu Devin. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016. 165, 214
- [3] Hamed Abdelhaq, Christian Sengstock, and Michael Gertz. Eventweet: Online localized event detection from twitter. *PVLDB*, 6(12):1326–1329, 2013. 106
- [4] Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan. Youtube-8m: A large-scale video classification benchmark. *arXiv preprint arXiv:1609.08675*, 2016. 155, 173
- [5] Alex Acero, Neal Bernstein, Rob Chambers, Yun-Cheng Ju, Xinggang Li, Julian Odell, Patrick Nguyen, Oliver Scholz, and Geoffrey Zweig. Live search for mobile: Web services by voice on the cellphone. In *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, pages 5256–5259. IEEE, IEEE, 2008. 151
- [6] Amina Adadi and Mohammed Berrada. Peeking inside the black-box: a survey on explainable artificial intelligence (xai). *IEEE access*, 6:52138–52160, 2018. 254
- [7] Stavros P Adam, Stamatios-Aggelos N Alexandropoulos, Panos M Pardalos, and Michael N Vrahatis. No free lunch theorem: A review. *Approximation and optimization*, pages 57–82, 2019. 58
- [8] Stergos D. Afantinos, Irene Doura, Eleni Kapellou, and Vangelis Karkaletsis. Exploiting cross-document relations for multi-document evolving summarization. *CoRR*, cs.CL/0404049, 2004. 119
- [9] Charu C Aggarwal. Data classification. In *Data Mining*, pages 285–344. Springer, 2015. 37
- [10] Charu C Aggarwal et al. Neural networks and deep learning. *Springer*, 10:978–3, 2018. 39
- [11] Charu C Aggarwal and Karthik Subbian. Event detection in social streams. In *SDM*, pages 624–635, 2012. 106

- [12] Fotis Aisopos, George Papadakis, Konstantinos Tserpes, and Theodora A. Varvarigou. Content vs. context for sentiment analysis: a comparative analysis over microblogs. In *ACM Conference on Hypertext and Social Media*, pages 187–196, 2012. 106
- [13] Ahmet Aker, Fabio Celli, Adam Funk, Emina Kurtic, Mark Hepple, and Rob Gaizauskas. Sheffield-trento system for sentiment and argument structure enhanced comment-to-article linking in the online news domain, 2016. 125, 126
- [14] Rasim M Alguliev, Ramiz M Aliguliyev, Makrufa S Hajirahimova, and Chingiz A Mehdiyev. Mcmr: Maximum coverage and minimum redundant text summarization model. *Expert Systems with Applications*, 38(12):14514–14522, 2011. 97
- [15] James Allan, Courtney Wade, and Alvaro Bolivar. Retrieval and novelty detection at the sentence level. In Charles L. A. Clarke, Gordon V. Cormack, Jamie Callan, David Hawking, and Alan F. Smeaton, editors, *SIGIR*, pages 314–321. ACM, 2003. 127
- [16] Berna Altınel and Murat Can Ganiz. Semantic text classification: A survey of past and recent advances. *Information Processing & Management*, 54(6):1129–1153, 2018. 184
- [17] Mohamed Aly. Survey on multiclass classification methods. *Neural Netw*, 19:1–9, 2005. 250
- [18] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*, 2016. 180
- [19] Leticia H Anaya. *Comparing Latent Dirichlet Allocation and Latent Semantic Analysis as Classifiers*. ERIC, 2011. 180
- [20] R. Angheluta, R. De Busser, and M. F. Moens. The use of topic segmentation for automatic summarization. In *Proceedings of the Document Understanding Conference (DUC)*, 2002. 132
- [21] Xavier Anguera, Simon Bozonnet, Nicholas Evans, Corinne Fredouille, Gerald Friedland, and Oriol Vinyals. Speaker diarization: A review of recent research. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(2):356–370, 2012. 37
- [22] Giorgos Argyriou, George Papadakis, George Stamoulis, Efi Karra Taniskidou, Niki-foros Pittaras, George Giannakopoulos, Sergio Albani, Michele Lazzarini, Emanuele Angiuli, Anca Popescu, Argyros Argyridis, and Manolis Koubarakis. Geosensor: Online scalable change and event detection over big data. In Pierre-Antoine Champin, Fabien L. Gandon, Mounia Lalmas, and Panagiotis G. Ipeirotis, editors, *WWW (Companion Volume)*, pages 223–226. ACM, 2018. 138
- [23] Abdelkrime Aries, Djamel Eddine Zegour, and Khaled Walid Hidouci. Allsummarizer system at multiling 2015: Multilingual single and multi-document summarization. In

Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 237–244. The Association for Computer Linguistics, 2015. 125, 235

- [24] Ebru Arisoy, Tara N Sainath, Brian Kingsbury, and Bhuvana Ramabhadran. Deep neural network language models. In *Proceedings of the NAACL-HLT 2012 Workshop: Will We Ever Really Replace the N-gram Model? On the Future of Language Modeling for HLT*, pages 20–28, 2012. 54
- [25] Leila Arras, Franziska Horn, Grégoire Montavon, Klaus-Robert Müller, and Wojciech Samek. ”what is relevant in a text document?”: An interpretable machine learning approach. *PLoS one*, 12(8):e0181142, 2017. 53
- [26] Farzindar Atefeh and Wael Khreich. A survey of techniques for event detection in twitter. *Computational Intelligence*, 31(1):132–164, 2015. 106, 119
- [27] Sören Auer, Simon Scerri, and Aad Versteden et. al. The bigdataeurope platform — supporting the variety dimension of big data. In *ICWE*, pages 41–59, 2017. 107
- [28] Nilar Aye, Fumio Hattori, and Kazuhiro Kuwabara. Use of ontologies for bridging semantic gaps in distant communication, 2008. 180
- [29] Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In Nicoletta Calzolari, Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, Mike Rosner, and Daniel Tapia, editors, *Proceedings of the International Conference on Language Resources and Evaluation*, 2010. 88, 181, 193
- [30] Pinkesh Badjatiya, Shashank Gupta, Manish Gupta, and Vasudeva Varma. Deep learning for hate speech detection in tweets. In *Proceedings of the 26th International Conference on World Wide Web Companion*, pages 759–760. International World Wide Web Conferences Steering Committee, 2017. 63, 64
- [31] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv1409.0473 [cs, stat]*, sep 2014. 92
- [32] Mingsian R Bai and Meng-chun Chen. Intelligent preprocessing and classification of audio signals. *Journal of the Audio Engineering Society*, 55(5):372–384, 2007. 38
- [33] Collin F Baker, Charles J Fillmore, and John B Lowe. The berkeley framenet project. In *Proceedings of the 17th international conference on Computational linguistics- Volume 1*, pages 86–90. Association for Computational Linguistics, Morgan Kaufmann Publishers / ACL, 1998. 181, 236
- [34] Alexandra Balahur and Hristo Tanev. Detecting implicit expressions of affect from text using semantic knowledge on common concept properties. In *LREC*, 2016. 124, 126

- [35] Babu Kaji Baniya, Joonwhoan Lee, and Ze-Nian Li. Audio feature reduction and analysis for automatic music genre classification. In *2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 457–462. IEEE, 2014. 47, 52
- [36] Elena Baralis, Aless Fiori, and ro. Summarizing biological literature with biosumm. In Jimmy Huang, Nick Koudas, Gareth J. F. Jones, Xindong Wu, Kevyn Collins-Thompson, and Aijun An, editors, *CIKM*, pages 1961–1962. ACM, 2010. 119
- [37] Marco Baroni, Georgiana Dinu, and Germán Kruszewski. Don’t count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 238–247, 2014. 54
- [38] John L Barron, David J Fleet, and Steven S Beauchemin. Performance of optical flow techniques. *International journal of computer vision*, 12(1):43–77, 1994. 153
- [39] Regina Barzilay and Kathleen R McKeown. Sentence fusion for multidocument news summarization. *Computational Linguistics*, 31(3):297–328, 2005. 37, 119
- [40] Michele Basile. Distance measures for signal processing and pattern recognition. *Signal processing*, 18(4):349–369, 1989. 48
- [41] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Computer vision and image understanding*, 110(3):346–359, 2008. 48
- [42] BBC News. Gaming worth more than video and music combined. <https://www.bbc.com/news/technology-46746593>, 2019. Accessed 26 January 2020. 85
- [43] Sören Becker, Marcel Ackermann, Sebastian Lapuschkin, Klaus-Robert Müller, and Wojciech Samek. Interpreting and explaining deep neural networks for classification of audio signals. *arXiv preprint arXiv:1807.03418*, 2018. 53
- [44] Kedar Bellare, Anish Das Sarma, Atish Das Sarma, Navneet Loival, Vaibhav Mehta, Ganesh Ramakrishnan, and Pushpak Bhattacharyya. Generic text summarization using wordnet. In *LREC*, pages 303–304. European Language Resources Association, 2004. 237
- [45] Richard Bellman. Dynamic programming and stochastic control processes. *Information and Control*, 1(3):228 – 239, 1958. 75
- [46] Richard Bellman. *Dynamic programming*. Courier Corporation, 2013. 50
- [47] Y. Bengio, A. Courville, and P. Vincent. Representation Learning: A Review and New Perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, August 2013. 39, 53, 56, 180
- [48] Yoshua Bengio. Deep learning of representations for unsupervised and transfer learning. In *JMLR Work. Conf. Proc.*, volume 7, pages 1–20. JMLR.org, jun 2011. 39

- [49] Yoshua Bengio. Practical recommendations for gradient-based training of deep architectures. In *Neural networks: Tricks of the trade*, pages 437–478. Springer, 2012. 53
- [50] Yoshua Bengio, Olivier Delalleau, and Nicolas Le Roux. The curse of dimensionality for local kernel machines. *Techn. Rep.*, 1258:12, 2005. 50
- [51] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. A neural probabilistic language model. *The journal of machine learning research*, 3:1137–1155, 2003. 54, 197, 200
- [52] Yoshua Bengio et al. Learning deep architectures for ai. *Foundations and trends® in Machine Learning*, 2(1):1–127, 2009. 38, 39, 46, 50, 52, 53, 198
- [53] Yoshua Bengio, Pascal Lamblin, Dan Popovici, and Hugo Larochelle. Greedy layer-wise training of deep networks. In *Advances in neural information processing systems*, pages 153–160, 2007. 53
- [54] Yoshua Bengio, Yann LeCun, et al. Scaling learning algorithms towards ai. *LARGE-scale kernel machines*, 34(5):1–41, 2007. 53
- [55] Ana B Benitez and Shih-Fu Chang. Image classification using multimedia knowledge networks. In *Proceedings 2003 International Conference on Image Processing (Cat. No. 03CH37429)*, volume 3, pages III–613. IEEE, 2003. 187, 189
- [56] Konstantina Bereta, Panayiotis Smeros, and Manolis Koubarakis. Representation and querying of valid time of triples in linked geospatial data. In *ESWC*, pages 259–274, 2013. 114
- [57] Dario Bertero and Pascale Fung. Deep learning of audio and language features for humor prediction. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 496–501, 2016. 187, 194
- [58] Siddhartha Bhattacharyya. A brief survey of color image preprocessing and segmentation techniques. 6:120–129, 2011. 38
- [59] Jiang Bian, Bin Gao, and Tie-Yan Liu. Knowledge-powered deep learning for word embedding. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 132–148. Springer, Springer, 2014. 201
- [60] Xiaoyong Bian, Chen Chen, Long Tian, and Qian Du. Fusing local and global features for high-resolution scene classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 10(6):2889–2901, 2017. 47, 49
- [61] Chris Biemann and Martin Riedl. Text: Now in 2d! a framework for lexical expansion with contextual similarity. *Journal of Language Modelling*, 1(1):55–95, 2013. 184

- [62] Alexander Binder, Motoaki Kawanabe, and Ulf Brefeld. Efficient classification of images with taxonomies. In *Asian Conference on Computer Vision*, pages 351–362. Springer, 2009. 187, 189
- [63] Steven Bird, Edward Loper, and Ewan Klein. *Natural Language Processing with Python*. O'Reilly Media Inc., 2009. 90, 91
- [64] Staffan Bjork and Jussi Holopainen. *Patterns in Game Design*. Charles River Media, 2004. 103
- [65] David M. Blei. Probabilistic topic models. *Communications of the ACM*, 55(4):77–84, 2012. 73, 87, 103
- [66] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003. 52, 54, 72, 87, 125, 133
- [67] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017. 235
- [68] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250. ACM, ACM, 2008. 181, 182, 236
- [69] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*, pages 2787–2795, 2013. 193
- [70] Andrea Borghesi, Federico Baldo, and Michela Milano. Improving deep learning models via constraint-based domain knowledge: a brief survey. *arXiv preprint arXiv:2005.10691*, 2020. 184
- [71] Anna Bosch, Andrew Zisserman, and Xavier Munoz. Image classification using random forests and ferns. In *2007 IEEE 11th international conference on computer vision*, pages 1–8. Ieee, 2007. 47, 51
- [72] Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186. Springer, 2010. 166
- [73] Peter Bourgonje, Julian Moreno-Schneider, Ankit Srivastava, and Georg Rehm. Automatic classification of abusive language and personal attacks in various forms of online communication. In *International Conference of the German Society for Computational Linguistics and Language Technology*, pages 180–191. Springer, 2017. 63, 64
- [74] Francesca Bovolo and Lorenzo Bruzzone. The time variable in data fusion: A change detection perspective. *IEEE Geosc. Remote Sensing Mag.*, 3(3):8–26, 2015. 104

- [75] Sing T Bow. *Pattern recognition and image preprocessing*. CRC press, 2002. 38
- [76] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 3(1):1–122, 2011. 201
- [77] Herrera Boyer, Xavier Serra, and Geoffroy Peeters. Audio descriptors and descriptor schemes in the context of mpeg-7. In *Proceedings of the 1999 International Computer Music Conference, ICMC; 1999 Oct 22-27; Beijing, China.[Michigan]: Michigan Publishing; 1999*. p. 581-4. International Computer Music Conference, 1999. 43, 49
- [78] Margaret M Bradley and Peter J Lang. Affective norms for english words (anew): Instruction manual and affective ratings. Technical report, Technical report C-1, the center for research in psychophysiology ..., 1999. 183
- [79] Darin Brezeale and Diane J Cook. Automatic video classification: A survey of the literature. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 38(3):416–430, 2008. 150
- [80] Sergey Brin and Lawrence Page. Reprint of: The anatomy of a large-scale hyper-textual web search engine. *Comput. Networks*, 56(18):3825–3833, 2012. 123
- [81] Alexander Brown. What is hate speech? part 1: The myth of hate. *Law and Philosophy*, 36(4):419–468, 2017. 62
- [82] Peter F Brown, Peter V DeSouza, Robert L Mercer, Vincent J Della Pietra, and Jenifer C Lai. Class-based n-gram models of natural language. *Comput. Linquistics*, 18(1950):467–479, 1992. 49, 54, 235
- [83] Grégoire Burel, Hassan Saif, and Harith Alani. Semantic wide and deep learning for detecting crisis-information categories on social media. In */ISWC*, pages 138–155, 2017. 106
- [84] Christopher JC Burges, John C Platt, and Soumya Jana. Extracting noise-robust features from audio data. In *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages I–1021. IEEE, 2002. 38
- [85] Nadia Burkart and Marco F Huber. A survey on the explainability of supervised machine learning. *Journal of Artificial Intelligence Research*, 70:245–317, 2021. 254
- [86] Marco Cadoli and Francesco M. Donini. A survey on knowledge compilation. *AI Commun.*, 10(3-4):137–150, 1997. 180
- [87] Xiaoyan Cai and Wenjie Li. Mutually reinforced manifold-ranking based relevance propagation model for query-focused multi-document summarization. *IEEE Trans. Speech Audio Process.*, 20(5):1597–1607, 2012. 124

- [88] Jose Camacho-Collados and Mohammad Taher Pilehvar. From word to sense embeddings: A survey on vector representations of meaning. *Journal of Artificial Intelligence Research*, 63:743–788, 2018. 184
- [89] Francesco Camastra and Alessandro Vinciarelli. *Machine learning for audio, image and video analysis: theory and applications*. Springer, 2015. 38
- [90] Pedro Cano, Markus Koppenberger, Perfecto Herrera, Sylvain Le Groux, Julien Richard, and Nicolas Wack. Nearest-neighbor generic sound classification with a wordnet-based taxonomy. In *Audio Engineering Society Convention 116*. Audio Engineering Society, 2004. 187, 190
- [91] Joan Capdevila, Gonzalo Pericacho, Jordi Torres, and Jesús Cerquides. Scaling dbscan-like algorithms for event detection systems in twitter. In *International Conference on Algorithms and Architectures for Parallel Processing*, pages 356–373. Springer, 2016. 136
- [92] Jaime G. Carbonell and Jade Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Research and Development in Information Retrieval*, pages 335–336, 1998. 126
- [93] Dallas Card, Chenhao Tan, and Noah A Smith. Neural models for documents with metadata. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 2031–2040, 2018. 201, 231
- [94] Joao Carreira, Rui Caseiro, Jorge Batista, and Cristian Sminchisescu. Free-form region description with second-order pooling. *IEEE transactions on pattern analysis and machine intelligence*, 37(6):1177–1189, 2014. 49
- [95] Sung-Hyuk Cha. Comprehensive survey on distance/similarity measures between probability density functions. *City*, 1(2):1, 2007. 48
- [96] Girish Chandrashekhar and Ferat Sahin. A survey on feature selection methods. *Computers & Electrical Engineering*, 40(1):16–28, 2014. 46
- [97] Simyung Chang, Hyoungwoo Park, Janghoon Cho, Hyunsin Park, Sungrack Yun, and Kyuwoong Hwang. Subspectral normalization for neural audio data processing. *arXiv preprint arXiv:2103.13620*, 2021. 38
- [98] Angelos Charalambidis, Antonis Troumpoukis, and Stasinos Konstantopoulos. Semagrow: Optimizing federated sparql queries. In *SEMANTiCS*, pages 121–128, 2015. 114
- [99] Chen Chen, Baochang Zhang, Hongjun Su, Wei Li, and Lu Wang. Land-use scene classification using multi-scale completed local binary patterns. *Signal, image and video processing*, 10(4):745–752, 2016. 52

- [100] Ping Chen, Soo Chin Liew, and Leong Keong Kwoh. Mangrove mapping and change detection using satellite imagery. In *IGARSS*, pages 5717–5720, 2017. 106
- [101] Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Diana Inkpen, and Si Wei. Neural natural language inference models enhanced with external knowledge. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2406–2417, 2018. 187, 193
- [102] Xinxiong Chen, Zhiyuan Liu, and Maosong Sun. A unified model for word sense representation and disambiguation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1025–1035, 2014. 125, 202, 207
- [103] Yanqing Chen, Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. The expressive power of word embeddings. *arXiv preprint arXiv:1301.3226*, 2013. 47, 54
- [104] Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng, and Jun Zhao. Event extraction via dynamic multi-pooling convolutional neural networks. In *ACL*, pages 167–176, 2015. 106
- [105] Xu Cheng, Cameron Dale, and Jiangchuan Liu. Statistics and social network of youtube videos. In *Quality of Service, 2008. IWQoS 2008. 16th International Workshop on*, pages 229–238. IEEE, 2008. 148
- [106] Davide Chicco. Siamese neural networks: An overview. *Artificial Neural Networks*, pages 73–94, 2021. 194
- [107] Keunwoo Choi, George Fazekas, and Mark Sandler. Explaining deep convolutional neural networks on music classification. *arXiv preprint arXiv:1607.02444*, 2016. 47, 56, 152
- [108] Keunwoo Choi, Gyorgy Fazekas, Mark Sandler, and Kyunghyun Cho. A comparison of audio signal preprocessing methods for deep neural networks on music tagging, 2018. 38
- [109] François Chollet et al. Keras. <https://keras.io>, 2015. 214
- [110] Sumit Chopra, Michael Auli, and Alexander M Rush. Abstractive sentence summarization with attentive recurrent neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 93–98, 2016. 125
- [111] Janara Christensen, Mausam, Soderl, Stephen , and Oren Etzioni. Towards coherent multi-document summarization. In Lucy Vanderwende, Hal Daumé III, and Katrin Kirchhoff, editors, *HLT-NAACL*, pages 1163–1173. The Association for Computational Linguistics, 2013. 123, 127

- [112] Xu Chu, Ihab F Ilyas, Sanjay Krishnan, and Jiannan Wang. Data cleaning: Overview and emerging challenges. In *Proceedings of the 2016 international conference on management of data*, pages 2201–2206, 2016. 38
- [113] Freddy Chua and Sitaram Asur. Automatic summarization of events from social media. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 7, 2013. 37
- [114] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014. 176, 233
- [115] Andrew P Clark, Kate L Howard, Andy T Woods, Ian S Penton-Voak, and Christof Neumann. Why rate when you could compare? using the “elochoice” package to assess pairwise comparisons of perceived physical strength. *PLoS one*, 13(1), 2018. 96
- [116] Guy Barrett Coleman and Harry C Andrews. Image segmentation by clustering. *Proceedings of the IEEE*, 67(5):773–785, 1979. 37
- [117] Allan M Collins and Elizabeth F Loftus. A spreading-activation theory of semantic processing. *Psychological review*, 82(6):407, 1975. 188, 207
- [118] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM, 2008. 47, 54, 197
- [119] Darrell Conklin. Melodic analysis with segment classes. *Machine Learning*, 65(2–3):349–360, 2006. 50
- [120] John Conroy and Sashka T Davis. Vector space models for scientific document summarization. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 186–191, 2015. 126
- [121] John M. Conroy and Hoa Trang Dang. Mind the gap: Dangers of divorcing evaluations of summary content from linguistic quality. In Donia Scott and Hans Uszkoreit, editors, *COLING*, pages 145–152, 2008. 119
- [122] John M. Conroy, Jeff Kubina, Peter A. Rankel, and Julia S. Yang. *Multilingual Summarization and Evaluation Using Wikipedia Featured Articles*, chapter Chapter 9, pages 281–336. 2015. 78, 237
- [123] John M. Conroy, Judith D. Schlesinger, and Dianne P. O’Leary. Classy 2009: Summarization and metrics. In *TAC*. NIST, 2009. 123, 127
- [124] John M. Conroy, Jade Goldstein Stewart, and Judith D. Schlesinger. Classy query-based multi-document summarization. In *In Proceedings of the Document*

Understanding Conf. Wksp. 2005 (DUC 2005) at the Human Language Technology Conf./Conf. on Empirical Methods in Natural Language Processing (HLT/EMNLP, 2005. 119, 122, 123, 127

- [125] Gregory W Corder and Dale I Foreman. *Nonparametric statistics: A step-by-step approach*. John Wiley & Sons, 2014. 188
- [126] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995. 64
- [127] George E Dahl, Dong Yu, Li Deng, and Alex Acero. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(1):30–42, 2012. 151
- [128] Joachim Daiber, Max Jakob, Chris Hokamp, and Pablo N. Mendes. Improving efficiency and accuracy in multilingual entity extraction. In *Proceedings of the 9th International Conference on Semantic Systems (I-Semantics)*, 2013. 182
- [129] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005. 51
- [130] CHENGHUA Dang and XINJUN Luo. Wordnet-based document summarization. In *WSEAS International Conference. Proceedings. Mathematics and Computers in Science and Engineering*, number 7. World Scientific and Engineering Academy and Society, 2008. 236, 237
- [131] Hoa Tran Dang. Overview of DUC 2006. In *Proc. Document Understanding Workshop*, page 10 pages. NIST, 2006. 121
- [132] Hoa Trang Dang. Overview of DUC 2005. In *Proceedings of the 2005 Document Understanding Workshop*, 2005. 121, 123
- [133] Hoa Trang Dang and Karolina Owczarzak. Overview of the tac 2008 update summarization task. In *Proceedings of the Text Analysis Conference*, 2008. 97, 119, 121
- [134] Marina Danilevsky, Kun Qian, Ranit Aharonov, Yannis Katsis, Ban Kawas, and Prithviraj Sen. A survey of the state of explainable ai for natural language processing. *arXiv preprint arXiv:2010.00711*, 2020. 46
- [135] Mittal C Darji and Dipti Mathpal. A review of video classification techniques. *IRJET Journal*, 4(6), 2017. 149
- [136] Kareem Darwish, Walid Magdy, and Tahar Zanouda. Trump vs. hillary: What went viral during the 2016 us presidential election. In *Proceedings of the International Conference on Social Informatics*, pages 143–161. Springer International Publishing, 2017. 87

- [137] Sanmay Das and Mike Y. Chen. Yahoo! for Amazon: extracting market sentiment from stock message boards. In *Proceedings of the Asia Pacific finance association annual conference*, 2001. 88
- [138] Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. Automated hate speech detection and the problem of offensive language. *arXiv preprint arXiv:1703.04009*, 2017. 63, 64, 65, 67
- [139] Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391, 1990. 51, 197, 235, 237
- [140] Fabio Del Vigna¹², Andrea Cimino²³, Felice Dell'Orletta, Marinella Petrocchi, and Maurizio Tesconi. Hate me, hate me not: Hate speech detection on facebook. 2017. 63, 64
- [141] Jonathan Delhumeau, Philippe-Henri Gosselin, Hervé Jégou, and Patrick Pérez. Revisiting the vlad image representation. In *Proceedings of the 21st ACM international conference on Multimedia*, pages 653–656. ACM, 2013. 51
- [142] Claudio Delli Bovi, Luis Espinosa-Anke, and Roberto Navigli. Knowledge base unification via sense embeddings and disambiguation. In *The 2015 Conference on Empirical Methods in Natural Language; 2015 Sept 17-21; Lisbon, Portugal.[Stroudsburg]: ACL (Association for Computational Linguistics); 2015. p. 726-36*. ACL (Association for Computational Linguistics), 2015. 184, 202
- [143] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009. 40, 181, 183
- [144] Li Deng, Ossama Abdel-Hamid, and Dong Yu. A deep convolutional neural network using heterogeneous pooling for trading acoustic invariance with phonetic confusion. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6669–6673. IEEE, 2013. 152
- [145] Pi-Sheng Deng. Inducing decision-making knowledge from data bases: An approach to automating knowledge acquisition. In Elias M. Awad, editor, *Proceedings of the 1990 ACM SIGBDP Conference on Trends and Directions in Expert Systems, October 31 - November 2, 1990, Orlando, FL, USA*, pages 189–211. ACM, 1990. 180
- [146] Thomas Deselaers and Vittorio Ferrari. Visual and semantic similarity in imagenet. In *CVPR*, pages 1777–1784. IEEE Computer Society, 2011. 187, 192
- [147] Sebastian Deterding, Dan Dixon, Rilla Khaled, and Lennart Nacke. From game design elements to gamefulness: Defining “gamification”. In *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments*, page 9–15, 2011. 85

- [148] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019. 55, 92, 193
- [149] Sander Dieleman and Benjamin Schrauwen. End-to-end learning for music audio. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 6964–6968. IEEE, 2014. 152
- [150] Nevenka Dimitrova, Lalitha Agnihotri, and Gang Wei. Video classification based on hmm using text and faces. In *Signal Processing Conference, 2000 10th European*, pages 1–4. IEEE, 2000. 153
- [151] Dandan Ding, Zhan Ma, Di Chen, Qingshuang Chen, Zoe Liu, and Fengqing Zhu. Advances in video compression system using deep neural network: a review and case studies. *Proceedings of the IEEE*, 2021. 148
- [152] Nemanja Djuric, Jing Zhou, Robin Morris, Mihajlo Grbovic, Vladan Radosavljevic, and Narayan Bhamidipati. Hate speech detection with comment embeddings. In *Proceedings of the 24th international conference on world wide web*, pages 29–30. ACM, 2015. 63, 64
- [153] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *International conference on machine learning*, pages 647–655. PMLR, 2014. 192
- [154] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2625–2634, 2015. 151, 153, 161, 164
- [155] Filip Karlo Došilović, Mario Brčić, and Nikica Hlupić. Explainable artificial intelligence: A survey. In *2018 41st International convention on information and communication technology, electronics and microelectronics (MIPRO)*, pages 0210–0215. IEEE, 2018. 46
- [156] H. P. Edmundson. New methods in automatic extracting. *J. ACM*, 16(2):264–285, 1969. 122
- [157] Cliff Edwards. Valve lines up console partners in challenge to microsoft, sony. <https://www.bloomberg.com/news/articles/2013-11-04/valve-lines-up-console-partners-in-challenge-to-microsoft-sony>, 2013. Accessed 26 January 2020. 88

- [158] Zakaria Elberrichi, Abdelattif Rahmoun, and Mohamed Amine Bentaalah. Using wordnet for text categorization. *International Arab Journal of Information Technology (IAJIT)*, 5(1), 2008. 187, 188, 200, 203, 204, 231, 237
- [159] Samira Ellouze, Maher Jaoua, and Lamia Hadrich Belguith. Machine learning approach to evaluate multilingual summaries. In George Giannakopoulos, Elena Lloret, John M. Conroy, Josef Steinberger, Marina Litvak, Peter A. Rankel, and Benoît Favre, editors, *MultiLing@EACL*, pages 47–54. Association for Computational Linguistics, 2017. 126
- [160] A. K. Elmagarmid, P. G. Ipeirotis, and V. S. Verykios. Duplicate record detection: A survey. *Knowledge and Data Engineering, IEEE Transactions on*, 19(1):1–16, 2007. 131
- [161] Jeffrey L Elman. Finding structure in time. *Cognitive science*, 14(2):179–211, 1990. 54
- [162] Eslam Elsawy, Moamen Mokhtar, and Walid Magdy. Tweetmogaz v2: Identifying news stories in social media. In *Proceedings of the 23rd ACM International Conference on Information and Knowledge Management*, 2014. 87
- [163] Entertainment Software Association. Essential facts about the computer and video game industry report. https://www.theesa.com/wp-content/uploads/2019/03/ESA_EssentialFacts_2018.pdf, 2018. Accessed: 5 Sep 2019. 85
- [164] Dumitru Erhan, Yoshua Bengio, Aaron Courville, Pierre-Antoine Manzagol, Pascal Vincent, and Samy Bengio. Why does unsupervised pre-training help deep learning? *Journal of Machine Learning Research*, 11(Feb):625–660, 2010. 54
- [165] Günes Erkan and Dragomir R. Radev. Lexpagerank: Prestige in multi-document text summarization. In *EMNLP*, pages 365–371. ACL, 2004. 119
- [166] Berna Erol, Dar-Shyang Lee, and Jonathan J. Hull. Multimodal summarization of meeting recordings. In *ICME*, pages 25–28. IEEE Computer Society, 2003. 119
- [167] Daniela Espinoza-Molina, Reza Bahmanyar, Ricardo Díaz-Delgado, Javier Bustamante, and Mihai Datcu. Land-cover change detection using local feature descriptors extracted from spectral indices. In *IGARSS*, pages 1938–1941, 2017. 106
- [168] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD*, pages 226–231, 1996. 110
- [169] David Kirk Evans, Judith L. Klavans, and Kathleen R. McKeown. Columbia news-blaster: Multilingual news summarization on the web. In *HLT-NAACL (Demonstration Papers)*. The Association for Computational Linguistics, 2004. 119, 121

- [170] Mark Everingham, Andrew Zisserman, Christopher KI Williams, Luc Van Gool, Moray Allan, Christopher M Bishop, Olivier Chapelle, Navneet Dalal, Thomas Deseilaers, and Gyuri Dorkó. The 2005 pascal visual object classes challenge. In *Machine Learning Challenges Workshop*, pages 117–176. Springer, 2005. 189
- [171] Angela Fan, David Grangier, and Michael Auli. Controllable abstractive summarization. In *Proceedings of the ACL Workshop on Neural Machine Translation and Generation*, 2017. 87
- [172] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008. 92
- [173] Stefano Faralli, Alexander Panchenko, Chris Biemann, and Simone P Ponzetto. Linked disambiguated distributional semantic networks. In *International Semantic Web Conference*, pages 56–64. Springer, 2016. 184
- [174] Manaal Faruqui, Jesse Dodge, Sujay Kumar Jauhar, Chris Dyer, Eduard Hovy, and Noah A Smith. Retrofitting word vectors to semantic lexicons. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1606–1615, 2015. 200, 203, 229, 244
- [175] Paolo Ferragina and Ugo Scaiella. Tagme: on-the-fly annotation of short text fragments (by wikipedia entities). In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 1625–1628, 2010. 193
- [176] Lorenzo Ferrone and Fabio Massimo Zanzotto. Symbolic, distributed, and distributional representations for natural language processing in the era of deep learning: A survey. *Frontiers in Robotics and AI*, 6:153, 2020. 184
- [177] Charles J Fillmore. Frame semantics. *Cognitive linguistics: Basic readings*, 34:373–400, 2006. 181
- [178] Charles J. Fillmore, Christopher R. Johnson, and Miriam R. L. Petrucc. Background to framenet. *International Journal of Lexicography*, 16(3):235–250, sep 2003. 181
- [179] Stefan Fischer, Rainer Lienhart, and Wolfgang Effelsberg. Automatic recognition of film genres. *Technical reports*, 95, 1995. 151, 153, 154
- [180] Evelyn Fix and Joseph L Hodges Jr. Discriminatory analysis-nonparametric discrimination: Small sample performance. Technical report, CALIFORNIA UNIV BERKELEY, 1952. 66
- [181] Lucie Flekova and Iryna Gurevych. Supersense embeddings: A unified model for supersense interpretation, prediction, and utilization. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2029–2041, 2016. 202, 229, 230

- [182] Jorge García Flores, Laurent Gillard, Olivier Ferret, and Gaël de Chalendar. Bag of senses versus bag of words: Comparing semantic and lexical approaches on sentence extraction. In *TAC*. NIST, 2008. 122
- [183] Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997. 191
- [184] Daniel Fried and Kevin Duh. Incorporating both distributional and relational semantics in word representations. *arXiv preprint arXiv:1412.4369*, 2014. 201
- [185] Zhouyu Fu, Guojun Lu, Kai Ming Ting, and Dengsheng Zhang. A survey of audio-based music classification and annotation. *IEEE transactions on multimedia*, 13(2):303–319, 2010. 43
- [186] Keinosuke Fukunaga. *Introduction to statistical pattern recognition*. Elsevier, 2013. 233
- [187] Sadaoki Furui. Cepstral analysis technique for automatic speaker verification. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 29(2):254–272, 1981. 167
- [188] Michel Galley, Jonathan Graehl, Kevin Knight, Daniel Marcu, Steve DeNeefe, Wei Wang, and Ignacio Thayer. Scalable inference and training of context-rich syntactic translation models. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 961–968. Association for Computational Linguistics, 2006. 235
- [189] Juri Ganitkevitch and Chris Callison-Burch. The multilingual paraphrase database. In *The 9th edition of the Language Resources and Evaluation Conference*, Reykjavik, Iceland, May 2014. European Language Resources Association. 183
- [190] Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. Ppdb: The paraphrase database. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 758–764, 2013. 40, 181, 183, 200
- [191] Huilin Gao, Lihua Dou, Wenjie Chen, and Jian Sun. Image classification with bag-of-words model based on improved SIFT algorithm. In *9th Asian Control Conference, ASCC 2013, Istanbul, Turkey, June 23-26, 2013*, pages 1–6. IEEE, 2013. 47, 49, 51
- [192] George Garbis, Kostis Kyzirakos, and Manolis Koubarakis. Geographica: A benchmark for geospatial RDF stores (long version). In *ISWC*, pages 343–359, 2013. 114
- [193] Salvador García, Julián Luengo, and Francisco Herrera. *Data preprocessing in data mining*, volume 72. Springer, 2015. 38

- [194] Jort F. Gemmeke, Daniel P. W. Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R. Channing Moore, Manoj Plakal, and Marvin Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *Proc. IEEE ICASSP 2017*, New Orleans, LA, 2017. 40, 152, 164, 165, 181, 183
- [195] Pierre-Etienne Genest, Guy Lapalme, and Mehdi Yousfi-Monod. Hextac: the creation of a manual extractive run. 2009. 76, 121
- [196] Shima Gerani, Giuseppe Carenini, and Raymond T. Ng. Modeling content and structure for abstractive review summarization. *Comput. Speech Lang.*, 53:302–331, 2019. 119
- [197] Allen Gersho and Robert M. Gray. Vector quantization and signal compression, 1992. 51, 52
- [198] Kranti Vithal Ghag and Ketan Shah. Comparative analysis of effect of stopwords removal on sentiment classification. In *2015 international conference on computer, communication and control (IC4)*, pages 1–6. IEEE, 2015. 38
- [199] George Giannakopoulos. *Automatic Summarization from Multiple Documents*. PhD thesis, University of the Aegean, 2009. 65, 130, 133
- [200] George Giannakopoulos, Mahmoud El-Haj, Benoît Favre, Marina Litvak, Josef Steinberger, and Vasudeva Varma. Tac2011 multiling pilot overview. In *TAC*. NIST, 2011. 119
- [201] George Giannakopoulos and Vangelis Karkaletsis. N-gram graphs: Representing documents and document sets in summary system evaluation. In *TAC*. NIST, 2009. 51, 119, 129
- [202] George Giannakopoulos and Vangelis Karkaletsis. Autosummeng and memog in evaluating guided summaries. In *TAC*. NIST, 2011. 131, 133
- [203] George Giannakopoulos and Vangelis Karkaletsis. Together we stand npower-ed. In *Proceedings of CICLING*, 2013. 129
- [204] George Giannakopoulos, Vangelis Karkaletsis, and George A Vouros. Testing the use of n-gram graphs in summarization sub-tasks. In *TAC*, 2008. 61
- [205] George Giannakopoulos, Vangelis Karkaletsis, George A. Vouros, and Panagiotis Stamatopoulos. Summarization system evaluation revisited: N-gram graphs. *ACM Trans. Speech Lang. Process.*, 5(3):5:1–5:39, 2008. 111, 112, 129, 130, 131
- [206] George Giannakopoulos, George Kiomourtzis, and Vangelis Karkaletsis. Newsum: “n-gram graph”-based summarization in the real world. In *Innovative Document Summarization Techniques: Revolutionizing Knowledge Understanding*, pages 205–230. IGI Global, 2014. 93, 111

- [207] George Giannakopoulos, Jeff Kubina, John Conroy, Josef Steinberger, Benoit Favre, Mijail Kabadjov, Udo Kruschwitz, and Massimo Poesio. MultiLing 2015: Multilingual Summarization of Single and Multi-Documents, On-line Fora, and Call-center Conversations. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 270–274, Prague, Czech Republic, 2015. Association for Computational Linguistics. 78, 96, 111, 237, 243
- [208] George Giannakopoulos, Petra Mavridi, Georgios Palioras, George Papadakis, and Konstantinos Tserpes. Representation models for text classification: a comparative analysis over three web document types. In *Proceedings of the 2nd international conference on web intelligence, mining and semantics*, page 13. ACM, 2012. 38, 47, 51, 65, 106, 111
- [209] George Giannakopoulos and Themis Palpanas. Adaptivity in entity subscription services. In *2009 Computation World: Future Computing, Service Computation, Cognitive, Adaptive, Content, Patterns*, pages 61–66. IEEE, 2009. 129
- [210] George Giannakopoulos, George A. Vouros, and Evangelis Karkaletsis. Mudos-ng: Multi-document summaries using n-gram graphs (tech report). *CoRR*, abs/1012.2042, 2010. 126, 129
- [211] Theodoros Giannakopoulos. pyaudioanalysis: An open-source python library for audio signal analysis. *PLoS one*, 10(12):e0144610, 2015. 166
- [212] Theodoros Giannakopoulos, Alexandros Makris, Dimitrios Kosmopoulos, Stavros Perantonis, and Sergios Theodoridis. Audio-visual fusion for detecting violent scenes in videos. In *Hellenic Conference on Artificial Intelligence*, pages 91–100. Springer, 2010. 154
- [213] Maria Giatsoglou, Manolis G Vozalis, Konstantinos Diamantaras, Athena Vakali, George Sarigiannidis, and Konstantinos Ch. Chatzisavvas Chatzisavvas. Sentiment analysis leveraging emotions and word embeddings. *Expert Systems with Applications*, 69:214–224, 2017. 101
- [214] Njagi Dennis Gitari, Zhang Zuping, Hanyurwimfura Damien, and Jun Long. A lexicon-based approach for hate speech detection. *International Journal of Multimedia and Ubiquitous Engineering*, 10(4):215–230, 2015. 63
- [215] Goran Glavaš and Ivan Vulić. Explicit retrofitting of distributional word vectors. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 34–45, 2018. 187, 191, 200
- [216] Josu Goikoetxea, Eneko Agirre, and Aitor Soroa. Single or multiple? combining word representations independently learned from text and wordnet. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pages 2608–2614. AAAI Press, 2016. 202

- [217] Jacob Goldberger, Geoffrey E Hinton, Sam Roweis, and Russ R Salakhutdinov. Neighbourhood components analysis. *Advances in neural information processing systems*, 17:513–520, 2004. 192
- [218] Maoguo Gong, Tao Zhan, Puzhao Zhang, and Qiguang Miao. Superpixel-based difference representation learning for change detection in multispectral remote sensing images. *IEEE Geosci. Remote Sensing*, 55(5):2658–2673, 2017. 106
- [219] Songjie Gong. A collaborative filtering recommendation algorithm based on user clustering and item clustering. *JSW*, 5(7):745–752, 2010. 37
- [220] Melvyn A Goodale and A David Milner. Separate visual pathways for perception and action. *Trends in neurosciences*, 15(1):20–25, 1992. 154
- [221] I Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT press, 2016. 149, 250
- [222] Allan D Gordon. A review of hierarchical classification. *Journal of the Royal Statistical Society: Series A (General)*, 150(2):119–137, 1987. 184
- [223] Kristen Grauman and Trevor Darrell. The pyramid match kernel: Discriminative classification with sets of image features. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 1458–1465. IEEE, 2005. 48
- [224] Derek Greene and Pádraig Cunningham. Practical solutions to the problem of diagonal dominance in kernel document clustering. In *Proceedings of the 23rd international conference on Machine learning*, pages 377–384. ACM, 2006. 229
- [225] Thomas Grill and Jan Schluter. Music boundary detection using neural networks on spectrograms and self-similarity lag matrices. In *Signal Processing Conference (EUSIPCO), 2015 23rd European*, pages 1296–1300. IEEE, 2015. 152
- [226] Miguel Grinberg. *Flask web development: developing web applications with python.* "O'Reilly Media, Inc.", 2018. 250
- [227] Jiuxiang Gu, Zhenhua Wang, Jason Kuen, Liyang Ma, Amir Shahroudy, Bing Shuai, Ting Liu, Xingxing Wang, Gang Wang, and Jianfei Cai. Recent advances in convolutional neural networks. *Pattern Recognition*, 77:354–377, 2018. 43
- [228] Jie Gui, Zhenan Sun, Shuiwang Ji, Dacheng Tao, and Tieniu Tan. Feature selection based on structured sparsity: A comprehensive study. *IEEE transactions on neural networks and learning systems*, 28(7):1490–1507, 2016. 46
- [229] Guodong Guo and Stan Z Li. Content-based audio classification and retrieval by support vector machines. *IEEE transactions on Neural Networks*, 14(1):209–215, 2003. 47, 49, 151

- [230] Ping Guo and Michael R Lyu. A study on color space selection for determining image segmentation region number. In *Proceedings of the International Conference on Artificial Intelligence*, volume 3, pages 1127–1132. Citeseer, 2000. 38
- [231] Wenzhong Guo, Jianwen Wang, and Shiping Wang. Deep multimodal representation learning: A survey. *IEEE Access*, 7:63373–63394, 2019. 149
- [232] Vishal Gupta and Gurpreet Singh Lehal. A Survey of Text Summarization Extractive Techniques. *Journal of Emerging Technologies in Web Intelligence*, 2(3), August 2010. 74, 235
- [233] Hongyu Han, Yongshi Zhang, Jianpei Zhang, Jing Yang, and Xiaomei Zou. Improving the performance of lexicon-based review sentiment analysis method by reducing additional introduced sentiment bias. *PLOS ONE*, 13, 2018. 88
- [234] Chris Harris and Mike Stephens. A combined corner and edge detector. In *Alvey vision conference*, volume 15, pages 10–5244. Citeseer, 1988. 48
- [235] Zellig S Harris. Distributional structure. *Word*, 10(2-3):146–162, 1954. 39, 54, 73, 235
- [236] Mohamad H Hassoun. *Fundamentals of artificial neural networks*. MIT press, 1995. 54
- [237] Trevor Hastie, Robert Tibshirani, Jerome Friedman, and James Franklin. The elements of statistical learning: data mining, inference and prediction. *The Mathematical Intelligencer*, 27(2):83–85, 2005. 235, 237
- [238] Haibo He and Yunqian Ma. Imbalanced learning: foundations, algorithms, and applications. 2013. 38
- [239] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 47, 55, 56, 176
- [240] Marti A. Hearst, Susan T Dumais, Edgar Osuna, John Platt, and Bernhard Scholkopf. Support vector machines. *IEEE Intelligent Systems and their applications*, 13(4):18–28, 1998. 48
- [241] Iris Hendrickx and Wauter Bosma. Using coreference links and sentence compression in graph-based summarization. In *TAC*. Citeseer, 2008. 124
- [242] Shawn Hershey, Sourish Chaudhuri, Daniel PW Ellis, Jort F Gemmeke, Aren Jansen, R Channing Moore, Manoj Plakal, Devin Platt, Rif A Saurous, and Bryan Seybold. Cnn architectures for large-scale audio classification. In *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, pages 131–135. IEEE, 2017. 47, 56, 152, 173

- [243] Ruben Hillewaere, Bernard Manderick, and Darrell Conklin. Global feature versus event models for folk song classification. In *ISMIR*, volume 2009, page 10th. Citeseer, 2009. 47, 49
- [244] Swapnil Hingmire, Sandeep Chougule, Girish K Palshikar, and Sutanu Chakraborti. Document classification by topic labeling. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pages 877–880. ACM, 2013. 197
- [245] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, and Tara N Sainath. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, 2012. 151
- [246] Geoffrey E Hinton. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800, 2002. 151
- [247] Geoffrey E Hinton. Deep belief networks. *Scholarpedia*, 4(5):5947, 2009. 56
- [248] Geoffrey E Hinton, James L McClelland, and David E Rumelhart. *Distributed representations*. Carnegie-Mellon University Pittsburgh, PA, 1984. 50, 197
- [249] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006. 53, 151
- [250] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006. 53, 176
- [251] Sepp Hochreiter. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(02):107–116, 1998. 55
- [252] Sepp Hochreiter and Jurgen Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1–32, 1997. 125, 159, 233, 244
- [253] Thomas Hofmann. Probabilistic latent semantic analysis. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pages 289–296. Morgan Kaufmann Publishers Inc., 1999. 51
- [254] Thomas Hofmann. Probabilistic latent semantic indexing. In *ACM SIGIR Forum*, volume 51, pages 211–218. ACM, 2017. 51
- [255] Frederik Hogenboom, Flavius Frasincar, Uzay Kaymak, and Franciska De Jong. An overview of event extraction from text. In *DeRiVE@ ISWC*, pages 48–57. Citeseer, 2011. 138
- [256] Berthold Horn, Berthold Klaus, and Paul Horn. *Robot vision*. MIT press, 1986. 153

- [257] Roger A Horn and Charles R Johnson. *Matrix analysis*. Cambridge university press, 2012. 235
- [258] Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, page 168–177, 2004. 86, 88, 102
- [259] Xiao Hu and J Stephen Downie. Improving mood classification in music digital libraries by combining lyrics and audio. In *Proceedings of the 10th annual joint conference on Digital libraries*, pages 159–168, 2010. 187, 189
- [260] Ya-Han Hu, Yen-Liang Chen, and Hui-Ling Chou. Opinion mining from online hotel reviews – a text summarization approach. *Information Processing & Management*, 53(2):436–449, 2017. 86, 87
- [261] Chunpeng Huang, Tianjun Fu, and Hsinchun Chen. Text-based video content classification for online video-sharing sites. *Journal of the American Society for Information Science and Technology*, 61(5):891–906, 2010. 153
- [262] Eric H Huang, Richard Socher, Christopher D Manning, and Andrew Y Ng. Improving word representations via global context and multiple word prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 873–882. Association for Computational Linguistics, 2012. 54, 202, 229, 230
- [263] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. 47, 55
- [264] Guang-Bin Huang, Hongming Zhou, Xiaojian Ding, and Rui Zhang. Extreme learning machine for regression and multiclass classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 42(2):513–529, 2011. 49
- [265] Xuedong D Huang, Yasuo Ariki, and Mervyn A Jack. Hidden markov models for speech recognition. 1990. 153
- [266] Zhengwei Huang, Ming Dong, Qirong Mao, and Yongzhao Zhan. Speech emotion recognition using cnn. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 801–804. ACM, 2014. 47, 56, 152
- [267] David H Hubel and Torsten N Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of physiology*, 160(1):106–154, 1962. 52, 56, 152
- [268] Chihli Hung and Stefan Wermter. Neural network based document clustering using wordnet ontologies. *International Journal of Hybrid Intelligent Systems*, 1(3-4):127–142, 2004. 204, 237

- [269] San-Yih Hwang, Chia-Yu Lai, Jia-Jhe Jiang, and Shanlin Chang. The identification of noteworthy hotel reviews for hotel management. *Pacific Asia Journal of the Association for Information Systems*, 6, 2014. 88
- [270] Ignacio Iacobacci, Mohammad Taher Pilehvar, and Roberto Navigli. Sensemed: Learning sense embeddings for word and relational similarity. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 95–105, 2015. 184, 202, 229, 230
- [271] Masaru Isonuma, Toru Fujino, Junichiro Mori, Yutaka Matsuo, and Ichiro Sakata. Extractive summarization using multi-task learning with document classification. In Martha Palmer, Rebecca Hwa, and Sebastian Riedel, editors, *EMNLP*, pages 2101–2110. Association for Computational Linguistics, 2017. 119
- [272] Giridharan Iyengar and Andrew B Lippman. Models for automatic classification of video sequences. In *Storage and Retrieval for Image and Video Databases VI*, volume 3312, pages 216–228. International Society for Optics and Photonics, 1997. 153
- [273] Mohit Iyyer, Varun Manjunatha, Jordan Boyd-Graber, and Hal Daumé III. Deep unordered composition rivals syntactic methods for text classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 1681–1691, 2015. 191
- [274] Jagadeesh J, Prasad Pingali, Vasudeva Varma, Jagadeesh J, Prasad Pingali, and Vasudeva Varma. *Sentence Extraction Based Single Document Summarization*. 2008. 74
- [275] Hajira Jabeen, Phil Archer, Simon Scerri, Aad Versteden, Ivan Ermilov, Giannis Mouchakis, Jens Lehmann, and Soeren Auer. Big data europe. In *EDBT/ICDT Workshops*, 2017. 138
- [276] Anil K. Jain and Richard C. Dubes. *Algorithms for Clustering Data*. Prentice-Hall, 1988. 37, 51
- [277] Mihir Jain, Herve Jegou, and Patrick Bouthemy. Better exploiting motion for better action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2555–2562, 2013. 153
- [278] Adit Jamdar, Jessica Abraham, Karishma Khanna, and Rahul Dubey. Emotion analysis of songs based on lyrical and audio features. *arXiv preprint arXiv:1506.05012*, 2015. 187, 190
- [279] Mario Jarmasz and Stan Szpakowicz. Roget’s thesaurus and semantic similarity. *Recent Advances in Natural Language Processing III: Selected Papers from RANLP*, 2003:111, 2004. 191

- [280] Adam Jatowt and Mitsuru Ishizuka. Temporal multi-page summarization. *Web Intell. Agent Syst.*, 4(2):163–180, 2006. 123
- [281] Tony Jebara, Risi Kondor, and Andrew G. Howard. Probability product kernels. *J. Mach. Learn. Res.*, 5:819–844, 2004. 52
- [282] Hervé Jégou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez. Aggregating local descriptors into a compact image representation. In *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pages 3304–3311. IEEE, 2010. 51
- [283] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3d convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):221–231, 2013. 150, 153
- [284] Mingyang Jiang, Yanchun Liang, Xiaoyue Feng, Xiaojing Fan, Zhili Pei, Yu Xue, and Renchu Guan. Text classification based on deep belief network and softmax regression. *Neural Computing and Applications*, 29(1):61–70, 2018. 232
- [285] Yu-Gang Jiang, Zuxuan Wu, Jinhui Tang, Zechao Li, Xiangyang Xue, and Shi-Fu Chang. Modeling multimodal clues in a hybrid deep learning framework for video classification. *IEEE Transactions on Multimedia*, 2018. 155, 173
- [286] Yu-Gang Jiang, Guangnan Ye, Shih-Fu Chang, Daniel Ellis, and Alexander C Loui. Consumer video understanding: A benchmark database and an evaluation of human and machine performance. In *Proceedings of the 1st ACM International Conference on Multimedia Retrieval*, page 29. ACM, 2011. 155, 165
- [287] Abelino Jiménez, Benjamin Elizalde, and Bhiksha Raj. Sound event classification using ontology-based neural networks. In *Proceedings of the Annual Conference on Neural Information Processing Systems*, 2018. 187, 194
- [288] Yongcheng Jing, Yezhou Yang, Zunlei Feng, Jingwen Ye, Yizhou Yu, and Mingli Song. Neural style transfer: A review. *IEEE transactions on visualization and computer graphics*, 26(11):3365–3385, 2019. 37
- [289] Anjali Ganesh Jivani et al. A comparative study of stemming algorithms. *Int. J. Comp. Tech. Appl.*, 2(6):1930–1938, 2011. 38, 235
- [290] Thorsten Joachims. Text categorization with support vector machines: Learning with many relevant features. *Machine learning: ECML-98*, pages 137–142, 1998. 47, 48, 197
- [291] Jean-Michel Jolion and Walter Kropatsch. *Graph based representations in pattern recognition*, volume 12. Springer Science & Business Media, 2012. 38
- [292] Ian Jolliffe. Principal Component Analysis. In Miodrag Lovric, editor, *International Encyclopedia of Statistical Science*, pages 1094–1096. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011. 50, 51, 82, 233, 235, 237

- [293] Karen Spärck Jones. A statistical interpretation of term specificity and its application in retrieval. page 9, 2004. 74
- [294] Karen Spärck Jones. Automatic summarising: The state of the art. *Inf. Process. Manag.*, 43(6):1449–1481, 2007. 128
- [295] Corinne Jørgensen, Alejandro Jaimes, Ana B Benitez, and Shih-Fu Chang. A conceptual framework and empirical research for classifying visual descriptors. *Journal of the American Society for Information Science and Technology*, 52(11):938–947, 2001. 43, 48
- [296] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of tricks for efficient text classification. 2016. 229, 230, 237, 249
- [297] David A Jurgens, Peter D Turney, Saif M Mohammad, and Keith J Holyoak. Semeval-2012 task 2: Measuring degrees of relational similarity. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, pages 356–364. Association for Computational Linguistics, 2012. 202
- [298] John S Justeson and Slava M Katz. Technical terminology: some linguistic properties and an algorithm for identification in text. *Natural language engineering*, 1(1):9–27, 1995. 51
- [299] Jesper Juul. *Half Real. Videogames between Real Rules and Fictional Worlds*. MIT Press, 2005. 103
- [300] Mijail A. Kabadjov, Martin Atkinson, Josef Steinberger, Ralf Steinberger, and Erik Van der Goot. Newsgist: A multilingual statistical news summarizer. In José L. Balázar, Francesco Bonchi, Aristides Gionis, and Michèle Sebag, editors, *ECML/PKDD* (3), volume 6323 of *Lecture Notes in Computer Science*, pages 591–594. Springer, 2010. 119
- [301] M Kalaiselvi Geetha and S Palanivel. Video classification and shot detection for video retrieval applications. *International Journal of Computational Intelligence Systems*, 2(1):39–50, 2009. 149
- [302] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3128–3137, 2015. 161
- [303] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014. 150

- [304] Slava Katz. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE transactions on acoustics, speech, and signal processing*, 35(3):400–401, 1987. 197, 235
- [305] Anna Kazantseva and Stan Szpakowicz. Summarizing short stories. *Computational Linguistics*, 36(1):71–109, 2010. 123
- [306] Pei Ke, Haozhe Ji, Siyang Liu, Xiaoyan Zhu, and Minlie Huang. Sentilare: Sentiment-aware language representation learning with linguistic knowledge. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6975–6988, Online, 2020. Association for Computational Linguistics. 187, 193
- [307] James Kennedy and Russell Eberhart. Particle swarm optimization. In *Proceedings of ICNN’95-International Conference on Neural Networks*, volume 4, pages 1942–1948. IEEE, 1995. 189
- [308] Salman Hameed Khan, Xuming He, Fatih Porikli, and Mohammed Bennamoun. Forest change detection in incomplete satellite images with deep neural networks. *IEEE Trans. Geoscience and Remote Sensing*, 55(9):5407–5423, 2017. 106
- [309] Hyoung-Gook Kim, Nicolas Moreau, and Thomas Sikora. *MPEG-7 audio and beyond: Audio content indexing and retrieval*. John Wiley & Sons, 2006. 49
- [310] Samuel Kim, Panayiotis Georgiou, and Shrikanth Narayanan. Latent acoustic topic models for unstructured audio classification. 1, 2012. 47, 52
- [311] Yoon Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014. 191
- [312] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 176, 250
- [313] Alexander Klaser, Marcin Marszałek, and Cordelia Schmid. A spatio-temporal descriptor based on 3d-gradients. In *BMVC 2008-19th British Machine Vision Conference*, pages 275–1. British Machine Vision Association, 2008. 152
- [314] Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)*, 46(5):604–632, sep 1999. 123
- [315] Tomas Kliegr, Krishna Chandramouli, Jan Nemrava, Vojtech Svatek, and Ebroul Izquierdo. Combining image captions and visual analysis for image concept classification. In *Proceedings of the 9th International Workshop on Multimedia Data Mining: held in conjunction with the ACM SIGKDD 2008*, pages 8–17, 2008. 187, 189
- [316] Elly C Knight, Sergio Poo Hernandez, Erin M Bayne, Vadim Bulitko, and Benjamin V Tucker. Pre-processing spectrogram parameters improve the accuracy of bioacoustic classification using convolutional neural networks. *Bioacoustics*, 29(3):337–355, 2020. 38

- [317] Teuvo Kohonen. The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480, 1990. 189
- [318] Nikolaos Kolitsas, Octavian-Eugen Ganea, and Thomas Hofmann. End-to-end neural entity linking. *arXiv preprint arXiv:1808.07699*, 2018. 193
- [319] Qiuxiang Kong, Yong Xu, Wenwu Wang, and Mark D Plumley. Audio set classification with attention model: A probabilistic perspective. *arXiv preprint arXiv:1711.00927*, 2017. 174
- [320] Ioannis Kontopoulos, George Giannakopoulos, and Iraklis Varlamis. Distributing n-gram graphs for classification. In *Advances in Databases and Information Systems*, pages 3–11. Springer, 2017. 135
- [321] P Krejzl, J Steinberger, T Hercig, and T Brychcín. Uwb participation in the multiling’s onforums task, 2016. 125, 126
- [322] Gunther Kress. *Multimodality: A social semiotic approach to contemporary communication*. Routledge, 2009. 149
- [323] Balaji Krishnapuram, Lawrence Carin, Mario AT Figueiredo, and Alexander J Hartemink. Sparse multinomial logistic regression: Fast algorithms and generalization bounds. *IEEE transactions on pattern analysis and machine intelligence*, 27(6):957–968, 2005. 51
- [324] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 47, 55, 56, 156
- [325] Hildegarde Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. Hmdb: a large video database for human motion recognition. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2556–2563. IEEE, 2011. 154
- [326] Anurag Kumar, Maksim Khadkevich, and Christian Fugen. Knowledge transfer from weakly labeled audio using convolutional neural network for sound events and scenes. *arXiv preprint arXiv:1711.01369*, 2017. 174
- [327] Ch Kumar, an, Prasad Pingali, and Vasudeva Varma. Estimating risk of picking a sentence for document summarization. In Alexander F. Gelbukh, editor, *CICLING*, volume 5449 of *Lecture Notes in Computer Science*, pages 571–581. Springer, 2009. 123
- [328] Rekhil M Kumar and K Sreekumar. A survey on image feature descriptors. *Int J Comput Sci Inf Technol*, 5:7668–7673, 2014. 48
- [329] Shamanth Kumar, Huan Liu, Sameep Mehta, and L Venkata Subramaniam. From tweets to events: Exploring a scalable solution for twitter streams. *arXiv preprint:1405.1392*, 2014. 106

- [330] Vipin Kumar and Sonajharia Minz. Mood classification of lyrics using sentiwordnet. In *2013 International Conference on Computer Communication and Informatics*, pages 1–5. IEEE, 2013. 187, 188
- [331] Irene Kwok and Yuzhou Wang. Locate the hate: Detecting tweets against blacks. In *AAAI*, 2013. 62, 63
- [332] Kostis Kyzirakos, Manos Karpathiotakis, and Manolis Koubarakis. Strabon: A semantic geospatial DBMS. In *ISWC*, pages 295–311, 2012. 114
- [333] Kostis Kyzirakos, Dimitrianos Savva, Ioannis Vlachopoulos, Alexandros Vasileiou, Nikolaos Karalis, Manolis Koubarakis, and Stefan Manegold. Geotriples: Transforming geospatial data into rdf graphs using r2rml and rml mappings. *Web Semantics: Science, Services and Agents on the World Wide Web*, 2018. 114
- [334] Kostis Kyzirakos, Ioannis Vlachopoulos, Dimitrianos Savva, Stefan Manegold, and Manolis Koubarakis. Geotriples: a tool for publishing geospatial data as rdf graphs using r2rml mappings. In *ISWC*, pages 393–396, 2014. 114
- [335] Alexandre Lacoste, Alexandra Luccioni, Victor Schmidt, and Thomas Dandres. Quantifying the carbon emissions of machine learning. *arXiv preprint arXiv:1910.09700*, 2019. 180
- [336] John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001. 194
- [337] Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. Recurrent convolutional neural networks for text classification. *Twenty-Ninth AAAI Conf. Artif. Intell.*, pages 2267–2273, 2015. 47, 54
- [338] Avinash Lakshman and Prashant Malik. Cassandra: a decentralized structured storage system. *Operating Systems Review*, 44(2):35–40, 2010. 139
- [339] Thomas K Landauer, Peter W. Foltz, and Darrell Laham. An introduction to latent semantic analysis. *Discourse Processes*, 25(2-3):259–284, January 1998. 82
- [340] Ken Lang. Newsweeder: Learning to filter netnews. In *Machine Learning Proceedings 1995*, pages 331–339. Elsevier, 1995. 213, 231
- [341] Ivan Laptev. On space-time interest points. *International journal of computer vision*, 64(2-3):107–123, 2005. 167
- [342] Ivan Laptev, Marcin Marszalek, Cordelia Schmid, and Benjamin Rozenfeld. Learning realistic human actions from movies. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008. 173
- [343] Leah S. Larkey, James Allan, Margaret E. Connell, Alvaro Bolivar, and Courtney Wade. Umass at trec 2002: Cross language and novelty tracks. In Ellen M. Voorhees and Lori P. Buckland, editors, *TREC*, volume 500-251 of *NIST Special Publication*. National Institute of Standards and Technology (NIST), 2002. 127

- [344] Cyril Laurier, Olivier Lartillot, Tuomas Eerola, and Petri Toiviainen. Exploring relationships between audio features and emotion in music. In *ESCOM 2009: 7th triennial conference of european society for the cognitive sciences of music*, 2009. 47, 49
- [345] Dawn Lawrie, W Bruce Croft, and Arnold Rosenberg. Finding topic words for hierarchical summarization. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 349–357. ACM, 2001. 235
- [346] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 2169–2178. IEEE, 2006. 49
- [347] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *International Conference on Machine Learning*, pages 1188–1196, 2014. 47, 55, 124, 125, 237
- [348] Yann LeCun, Yoshua Bengio, Geoffrey Hinton, Lecun Y., Bengio Y., and Hinton G. Deep learning. *Nature*, 521(7553):436–444, may 2015. 234
- [349] Yulia Ledeneva. Effect of preprocessing on extractive summarization with maximal frequent sequences. In Alexander F. Gelbukh and Eduardo F. Morales, editors, *MICAI*, volume 5317 of *Lecture Notes in Computer Science*, pages 123–132. Springer, 2008. 123
- [350] Honglak Lee, Peter T Pham, Yan Largman, and Andrew Y Ng. Unsupervised feature learning for audio classification using convolutional deep belief networks. *Nips*, 9:1096–1104, 2009. 47, 56, 151
- [351] Keansub Lee and Daniel PW Ellis. Audio-based semantic concept classification for consumer video. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(6):1406–1416, 2010. 47, 52, 151
- [352] Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. End-to-end neural coreference resolution. *arXiv preprint arXiv:1707.07045*, 2017. 193
- [353] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, and Sören Auer. Dbpedia—a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, 6(2):167–195, 2015. 181, 182
- [354] D. S. Leite, L. H. M. Rino, T. A. S. Pardo, and M. G. V. Nunes. Extractive automatic summarization: Does more linguistic knowledge make a difference? In *Proceedings of the TextGraphs-2 HLT/NAACL Workshop*, Rochester, USA, April 2007. 123
- [355] Douglas B Lenat, Ramanathan V. Guha, Karen Pittman, Dexter Pratt, and Mary Shepherd. Cyc: toward programs with common sense. *Communications of the ACM*, 33(8):30–49, 1990. 181, 182

- [356] Leena Lepistoe, Iivari Kunttu, and Ari Visa. Rock image classification using color features in gabor space. *J. Electronic Imaging*, 14(4):040503, 2005. 47, 49
- [357] David D Lewis. Naive (bayes) at forty: The independence assumption in information retrieval. In *European conference on machine learning*, pages 4–15. Springer, 1998. 66
- [358] David D Lewis and Marc Ringuette. A comparison of two learning algorithms for text categorization. In *Third annual symposium on document analysis and information retrieval*, volume 33, pages 81–93, 1994. 50
- [359] Lei Li, Yazhao Zhang, Liyuan Mao, Junqi Chi, Moye Chen, and Zuying Huang. Cist@ clscisumm-17: Multiple features based citation linkage, classification and summarization. In *BIRNDL@ SIGIR* (2), pages 43–54, 2017. 125, 236
- [360] Min Li, Jian Tan, Yandong Wang, Li Zhang, and Valentina Salapura. Sparkbench: a comprehensive benchmarking suite for in memory data analytic platform spark. In *Proceedings of the 12th ACM international conference on computing frontiers*, pages 1–8, 2015. 136
- [361] Wei Li and Maosong Sun. Automatic image annotation based on wordnet and hierarchical ensembles. In Alexander F. Gelbukh, editor, *CICLing*, volume 3878 of *Lecture Notes in Computer Science*, pages 417–428. Springer, 2006. 187, 191
- [362] Wen Li, Li Niu, and Dong Xu. Exploiting privileged information from web data for image categorization. In *European Conference on Computer Vision*, pages 437–452. Springer, 2014. 187, 191
- [363] Yang Li, Quan Pan, Tao Yang, Suhang Wang, Jiliang Tang, and Erik Cambria. Learning word representations for sentiment analysis. *Cognitive Computation*, 9(6):843–851, 2017. 187, 190
- [364] Yiming Li, Baogang Wei, Liang Yao, Hui Chen, and Zherong Li. Knowledge-based document embedding for cross-domain text classification. In *Neural Networks (IJCNN), 2017 International Joint Conference on*, pages 1395–1402. IEEE, 2017. 201, 233
- [365] Yong H Li and Anil K Jain. Classification of text documents. *The Computer Journal*, 41(8):537–546, 1998. 47, 48, 50
- [366] Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. Gated graph sequence neural networks. *arXiv preprint arXiv:1511.05493*, 2015. 193
- [367] Antonios Liapis, Georgios N. Yannakakis, Mark J. Nelson, Mike Preuss, and Rafael Bidarra. Orchestrating game generation. *IEEE Transactions on Games*, 11(1):48–68, 2019. 85, 90
- [368] Andy Liaw, Matthew Wiener, et al. Classification and regression by randomforest. *R news*, 2(3):18–22, 2002. 66

- [369] Richard Liaw, Eric Liang, Robert Nishihara, Philipp Moritz, Joseph E Gonzalez, and Ion Stoica. Tune: A research platform for distributed model selection and training. *arXiv preprint arXiv:1807.05118*, 2018. 250
- [370] Thomas Lidy and Andreas Rauber. Evaluation of feature extractors and psycho-acoustic transformations for music genre classification. In *ISMIR*, pages 34–41, 2005. 47, 49
- [371] Antonio Lieto, Daniele Radicioni, Valentina Rho, and Enrico Mensa. Towards a unifying framework for conceptual representation and reasoning in cognitive systems. *Intelligenza Artificiale*, 11(2):139–153, 2017. 184
- [372] Kart-Leong Lim and Hamed Kiani Galoogahi. Shape classification using local and global features. In *Image and Video Technology (PSIVT), 2010 Fourth Pacific-Rim Symposium on*, pages 115–120. IEEE, 2010. 48
- [373] Chin-Yew Lin. ROUGE: A Package for Automatic Evaluation of Summaries. page 8, 2004. 79
- [374] Shih-Hsiang Lin and Berlin Chen. The ntu summarization system at tac 2009. In *TAC*. Citeseer, 2009. 124
- [375] Dimitri A Lisin, Marwan A Mattar, Matthew B Blaschko, Erik G Learned-Miller, and Mark C Benfield. Combining local and global image features for object class recognition. In *Computer vision and pattern recognition-workshops, 2005. CVPR workshops. IEEE Computer society conference on*, pages 47–47. IEEE, 2005. 48
- [376] Marina Litvak and Mark Last. Multilingual single-document summarization with muse. In *Proceedings of the MultiLing 2013 Workshop on Multilingual Multi-document Summarization*, pages 77–81, 2013. 124, 235
- [377] Marina Litvak, Mark Last, and Menahem Friedman. A new approach to improving multilingual summarization using a genetic algorithm. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 927–936, 2010. 124
- [378] Marina Litvak and Natalia Vanetik. Query-based summarization using mdl principle. In George Giannakopoulos, Elena Lloret, John M. Conroy, Josef Steinberger, Marina Litvak, Peter A. Rankel, and Benoît Favre, editors, *MultiLing@EACL*, pages 22–31. Association for Computational Linguistics, 2017. 125
- [379] Marina Litvak, Natalia Vanetik, Mark Last, and Elena Churkin. Museec: A multilingual text summarization tool. In Sameer Pradhan and Marianna Apidianaki, editors, *ACL (System Demonstrations)*, pages 73–78. Association for Computational Linguistics, 2016. 124
- [380] Hugo Liu and Push Singh. Conceptnet — a practical commonsense reasoning toolkit. *BT technology journal*, 22(4):211–226, 2004. 181, 182

- [381] Tao Liu, Zheng Chen, Benyu Zhang, Wei-ying Ma, and Gongyi Wu. Improving text classification using local latent semantic indexing. In *Data Mining, 2004. ICDM'04. Fourth IEEE International Conference on*, pages 162–169. IEEE, 2004. 47, 51
- [382] Yang Liu, Zhiyuan Liu, Tat-Seng Chua, and Maosong Sun. Topical word embeddings. In *Twenty-ninth AAAI conference on artificial intelligence*. Citeseer, 2015. 47, 54
- [383] Ying Liu, Peter Scheuermann, Xingsen Li, and Xingquan Zhu. Using wordnet to disambiguate word senses for text classification. In *International Conference on Computational Science*, pages 781–789. Springer, 2007. 204, 237
- [384] Ying Liu, Dengsheng Zhang, Guojun Lu, and Wei-Ying Ma. A survey of content-based image retrieval with high-level semantics. *Pattern recognition*, 40(1):262–282, 2007. 48
- [385] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019. 193
- [386] Zhu Liu, Jincheng Huang, and Yao Wang. Classification tv programs based on audio information using hidden markov model. In *Multimedia Signal Processing, 1998 IEEE Second Workshop on*, pages 27–32. IEEE, 1998. 151
- [387] Zhu Liu, Yao Wang, and Tsuhan Chen. Audio feature extraction and analysis for scene segmentation and classification. *Journal of VLSI signal processing systems for signal, image and video technology*, 20(1-2):61–79, 1998. 151
- [388] Elena Lloret, Ester Boldrini, Patricio Martínez-Barco, and Manuel Palomar. Ultra-concise multi-genre summarisation of web2. 0: towards intelligent content generation. In *Proceedings of the MultiLing 2017 Workshop on Summarization and Summary Evaluation Across Source Types and Genres*, pages 37–46, 2017. 125
- [389] Elena Lloret and Manuel Palomar. A gradual combination of features for building automatic summarisation systems. In *International Conference on Text, Speech and Dialogue*, pages 16–23. Springer, 2009. 235
- [390] Elena Lloret, Laura Plaza, and Ahmet Aker. The challenging task of summary evaluation: an overview. *Lang. Resour. Evaluation*, 52(1):101–148, 2018. 119
- [391] Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137, 1982. 237
- [392] Xianzhong Long, Hongtao Lu, Yong Peng, Xianzhong Wang, and Shaokun Feng. Image classification based on improved vlad. 75:5533–5555, 2016. 47, 51
- [393] Moshe Looks, Marcello Herreshoff, DeLesley Hutchins, and Peter Norvig. Deep learning with dynamic computation graphs. *arXiv preprint arXiv:1702.02181*, 2017. 191

- [394] Edward Loper and Steven Bird. Nltk: the natural language toolkit. In *Proceedings of the ACL-02 Workshop on Effective tools and methodologies for teaching natural language processing and computational linguistics-Volume 1*, pages 63–70. Association for Computational Linguistics, 2002. 214, 250
- [395] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004. 48, 167
- [396] Dengsheng Lu, P Mausel, E Brondizio, and Emilio Moran. Change detection techniques. *International journal of remote sensing*, 25(12):2365–2401, 2004. 103
- [397] Dengsheng Lu and Qihao Weng. A survey of image classification methods and techniques for improving classification performance. *International journal of Remote sensing*, 28(5):823–870, 2007. 43
- [398] Jiasen Lu, Caiming Xiong, Devi Parikh, and Richard Socher. Knowing when to look: Adaptive attention via a visual sentinel for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 6, page 2, 2017. 161
- [399] Lie Lu, Hong-Jiang Zhang, and Hao Jiang. Content analysis for audio classification and segmentation. *IEEE Transactions on speech and audio processing*, 10(7):504–516, 2002. 47, 49
- [400] Lie Lu, Hong-Jiang Zhang, and Stan Z Li. Content-based audio classification and segmentation by using support vector machines. *Multimedia systems*, 8(6):482–492, 2003. 47, 49
- [401] Hans Peter Luhn. The automatic creation of literature abstracts. *IBM J. Res. Dev.*, 2(2):159–165, 1958. 119
- [402] Jiebo Luo and Andreas Savakis. Indoor vs outdoor classification of consumer photographs using low-level and semantic features. In *Proceedings 2001 International Conference on Image Processing (Cat. No. 01CH37205)*, volume 2, pages 745–748. IEEE, 2001. 187, 189
- [403] Thang Luong, Richard Socher, and Christopher Manning. Better word representations with recursive neural networks for morphology. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 104–113, 2013. 201
- [404] Andrew L Maas and Andrew Y Ng. A probabilistic model for semantic word vectors. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, pages 1–8. ACM, 2010. 47, 54
- [405] Walid Magdy and Tamer Elsayed. Unsupervised adaptive microblog filtering for broad dynamic topics. *Information Processing & Management*, 52(4):513–528, 2016. 87, 101

- [406] Juha Makkonen, Helena Ahonen-Myka, and Marko Salmenkivi. Applying semantic classes in event detection and tracking. In *ICON 2002*, pages 175–183, 2002. 106
- [407] Juha Makkonen, Helena Ahonen-Myka, and Marko Salmenkivi. Simple semantics in topic detection and tracking. *Inf. Retr.*, 7(3-4):347–368, 2004. 106
- [408] Inderjeet Mani. Summarization evaluation: An overview. 2001. 39
- [409] Inderjeet Mani, Eric Bloedorn, and Barbara Gates. Using cohesion and coherence models for text summarization. In Eduard Hovy and Dragomir R. Radev, editors, *Proceedings of the Spring Symposium on Intelligent Text Summarization (AAAI 98)*, pages 69–76, Stanford, CA, March 1998. AAAI Press. 120, 122
- [410] Bangalore S Manjunath and Wei-Ying Ma. Texture features for browsing and retrieval of image data. *IEEE Transactions on pattern analysis and machine intelligence*, 18(8):837–842, 1996. 48
- [411] William C. Mann, S Thompson, and ra A. Rhetorical structure theory: A theory of text organization. In Livia Polanyi, editor, *The Structure of Discourse*. Ablex Publishing Corporation, Norwood, N.J., 1987. 122
- [412] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Scoring, term weighting, and the vector space model*, page 100–123. Cambridge University Press, 2008. 188
- [413] Trevor N Mansuy and Robert J Hilderman. Evaluating wordnet features in text classification models. In *FLAIRS Conference*, pages 568–573, 2006. 200, 203, 236
- [414] Kenneth Marino, Ruslan Salakhutdinov, and Abhinav Gupta. The more you know: Using knowledge graphs for image classification. *arXiv preprint arXiv:1612.04844*, 2016. 187, 193
- [415] Foteini Markatopoulou, Nikiforos Pittaras, Olga Papadopoulou, Vasileios Mezaris, and Ioannis Patras. A study on the use of a binary local descriptor and color extensions of local descriptors for video concept detection. In *International Conference on Multimedia Modeling*, pages 282–293. Springer, 2015. 48
- [416] Marcin Marszalek and Cordelia Schmid. Semantic hierarchies for visual object recognition. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–7. IEEE, 2007. 187, 189
- [417] James H Martin and Daniel Jurafsky. *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition*. Pearson/Prentice Hall Upper Saddle River, 2009. 205, 206
- [418] Sanda Martinčić-Ipšić, Tanja Miličić, and Ljupčo Todorovski. The influence of feature representation of text on the performance of document classification. *Applied Sciences*, 9(4):743, 2019. 38

- [419] Yutaka Matsuo, Yukio Ohsawa, and Mitsuru Ishizuka. A document as a small world. In Takao Terano, Toyoaki Nishida, Akira Namatame, Shusaku Tsumoto, Yukio Ohsawa, and Takashi Washio, editors, *JSAI Workshops*, volume 2253 of *Lecture Notes in Computer Science*, pages 444–448. Springer, 2001. 124
- [420] Jon D McAuliffe and David M Blei. Supervised topic models. In *Advances in neural information processing systems*, pages 121–128, 2008. 201
- [421] Andrew McCallum, Ronald Rosenfeld, Tom M Mitchell, and Andrew Y Ng. Improving text classification by shrinkage in a hierarchy of classes. In *ICML*, volume 98, pages 359–367, 1998. 191
- [422] Andrew Kachites McCallum. MALLET: A Machine Learning for Language Toolkit. 2002. 79
- [423] Victoria McCargar. Statistical approaches to automatic text summarization. *Bulletin of the American Society for Information Science and Technology*, 30(4):21–25, 2004. 235
- [424] Peter McCullagh and John A Nelder. *Generalized linear models*, volume 37. CRC press, 1989. 63
- [425] Gary McKeown, Michel F Valstar, Roderick Cowie, and Maja Pantic. The semaine corpus of emotionally coloured character interactions. In *Multimedia and Expo (ICME), 2010 IEEE International Conference on*, pages 1079–1084. IEEE, 2010. 154
- [426] Martin F. McKinney and Jeroen Breebaart. Features for audio and music classification. In *ISMIR 2003, 4th International Conference on Music Information Retrieval, Baltimore, Maryland, USA, October 27-30, 2003, Proceedings*, 2003. 47, 49
- [427] Larry R Medsker and LC Jain. Recurrent neural networks. *Design and Applications*, 5, 2001. 43
- [428] Rajiv Mehrotra, Kameswara Rao Namuduri, and Nagarajan Ranganathan. Gabor filter-based edge detection. *Pattern recognition*, 25(12):1479–1494, 1992. 49
- [429] Scott W Menard. *Applied logistic regression analysis*. Number 04; e-book. 1995. 66
- [430] Dirk Merkel. Docker: Lightweight linux containers for consistent development and deployment. *Linux J.*, 2014(239), mar 2014. 139
- [431] Nima Mesgarani, Malcolm Slaney, and Shihab A Shamma. Discrimination of speech from nonspeech based on multiscale spectro-temporal modulations. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(3):920–930, 2006. 38
- [432] Antoine Miech, Ivan Laptev, and Josef Sivic. Learnable pooling with context gating for video classification. *arXiv preprint arXiv:1706.06905*, 2017. 155

- [433] Pauli Miettinen. Matrix decomposition methods for data mining: Computational complexity and algorithms. 2009. 43
- [434] Rada Mihalcea and Paul Tarau. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 404–411, 2004. 123, 125, 235
- [435] Krystian Mikolajczyk, Bastian Leibe, and Bernt Schiele. Local features for object class recognition. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 2, pages 1792–1799. IEEE, 2005. 46
- [436] Krystian Mikolajczyk and Cordelia Schmid. An affine invariant interest point detector. In *European conference on computer vision*, pages 128–142. Springer, 2002. 48
- [437] Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, 2013. 101, 198, 235
- [438] Tomáš Mikolov, Martin Karafiat, Lukáš Burget, Jan Černocký, and Sanjeev Khudanpur. Recurrent neural network based language model. In *Eleventh Annual Conference of the International Speech Communication Association*, pages 1045–1048. ISCA, 2010. 197, 200
- [439] Tomas Mikolov and S Kombrink. Extensions of recurrent neural network language model. *Icassp*, pages 5528–5531, 2011. 197
- [440] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119. Curran Associates, Inc., 2013. 53, 54, 55, 188, 191, 198, 200, 202, 203, 235, 236, 249
- [441] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995. 40, 125, 181, 188, 200, 204, 236, 237, 249
- [442] George A Miller, Claudia Leacock, Randee Tengi, and Ross T Bunker. A semantic concordance. In *Proceedings of the workshop on Human Language Technology*, pages 303–308. Association for Computational Linguistics, 1993. 206
- [443] Shervin Minaee, Nal Kalchbrenner, Erik Cambria, Narjes Nikzad, Meysam Chenaghlou, and Jianfeng Gao. Deep learning-based text classification: A comprehensive review. *ACM Comput. Surv.*, 54(3), April 2021. 43
- [444] Vikramjit Mitra, Chia-Jiu Wang, and Satarupa Banerjee. Text classification: A least square support vector machine approach. *Applied Soft Computing*, 7(3):908–914, 2007. 47, 51

- [445] Andriy Mnih and Geoffrey E Hinton. A scalable hierarchical distributed language model. In *Advances in neural information processing systems*, pages 1081–1088, 2009. 54
- [446] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2018. 39
- [447] Simon Moncrieff, Svetha Venkatesh, and Chitra Dorai. Horror film genre typing and scene labeling via audio analysis. In *Multimedia and Expo, 2003. ICME'03. Proceedings. 2003 International Conference on*, volume 2, pages II–193. IEEE, 2003. 153
- [448] Tatsunori Mori. Information gain ratio as term weight: the case of summarization of ir results. In *COLING 2002: The 19th International Conference on Computational Linguistics*, 2002. 235
- [449] Frederic Morin and Yoshua Bengio. Hierarchical probabilistic neural network language model. In *Aistats*, volume 5, pages 246–252. Citeseer, 2005. 200, 204, 237
- [450] Andrea Moro, Alessandro Raganato, and Roberto Navigli. Entity linking meets word sense disambiguation: a unified approach. *Transactions of the Association for Computational Linguistics*, 2:231–244, 2014. 202
- [451] Hamdy Mubarak, Kareem Darwish, and Walid Magdy. Abusive language detection on arabic social media. In *Proceedings of the First Workshop on Abusive Language Online*, pages 52–56, 2017. 62, 63
- [452] Subhabrata Mukherjee and Pushpak Bhattacharyya. Feature specific sentiment analysis for product reviews. In *Computational Linguistics and Intelligent Text Processing*, pages 475–487. Springer Berlin Heidelberg, 2012. 86
- [453] Markus Nagel, Thomas Mensink, and Cees GM Snoek. Event fisher vectors: Robust encoding visual diversity of visual streams. In *BMVC*, volume 2, page 6, 2015. 173
- [454] Roberto Navigli. Word sense disambiguation: A survey. *ACM computing surveys (CSUR)*, 41(2):10, 2009. 205, 206
- [455] Roberto Navigli, Kenneth C Litkowski, and Orin Hargraves. Semeval-2007 task 07: Coarse-grained english all-words task. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 30–35, 2007. 202
- [456] Roberto Navigli and Simone Paolo Ponzetto. Babelnet: Building a very large multilingual semantic network. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 216–225. Association for Computational Linguistics, 2010. 181
- [457] Roberto Navigli and Simone Paolo Ponzetto. Babelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250, 2012. 202

- [458] Roberto Navigli and Paola Velardi. Learning domain ontologies from document warehouses and dedicated web sites. *Computational Linguistics*, 30(2):151–179, 2004. 103
- [459] Radford M Neal. Connectionist learning of belief networks. *Artificial intelligence*, 56(1):71–113, 1992. 151
- [460] Ani Nenkova and Lucy Vanderwende. The impact of frequency on summarization. *Microsoft Research, Redmond, Washington, Tech. Rep. MSR-TR-2005*, 101, 2005. 235
- [461] Joel Larocca Neto, Alex A Freitas, and Celso AA Kaestner. Automatic text summarization using a machine learning approach. In *Brazilian Symposium on Artificial Intelligence*, pages 205–215. Springer, 2002. 235
- [462] H Nezreg, H Lehbab, and H Belbachir. Conceptual representation using wordnet for text categorization. *International Journal of Computer and Communication Engineering*, 3(1):27, 2014. 187, 188, 200, 203, 231
- [463] Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Y Ng. Multimodal deep learning. *Proc. 28th Int. Conf. Mach. Learn.*, pages 689–696, 2011. 150
- [464] Thanh-Son Nguyen, Hady Wirawan Lauw, and Panayiotis Tsaparas. Review synthesis for micro-review summarization. In Xueqi Cheng, Hang Li, Evgeniy Gabrilovich, and Jie Tang, editors, *WSDM*, pages 169–178. ACM, 2015. 119
- [465] Thien Huu Nguyen and Ralph Grishman. Event detection and domain adaptation with convolutional neural networks. In *ACL*, pages 365–371, 2015. 106
- [466] John Niekrasz, Matthew Purver, John Dowding, and Stanley Peters. Ontology-based discourse understanding for a persistent meeting assistant. In *AAAI Spring Symposium: Persistent Assistants: Living and Working with AI*, pages 26–33. AAAI, 2005. 119
- [467] Michael A. Nielsen. Neural networks and deep learning, 2015. 39
- [468] Charalampos Nikolaou, Kallirroi Dogani, Konstantina Bereta, George Garbis, Manos Karpathiotakis, Kostis Kyriakos, and Manolis Koubarakis. Sextant: Visualizing time-evolving linked geospatial data. *J. Web Sem.*, 35:35–52, 2015. 114
- [469] Chikashi Nobata and Satoshi Sekine. Crl/nyu summarization system at duc-2004. In *Document Understanding Workshop 2004*, 2004. 87
- [470] Chikashi Nobata, Joel Tetreault, Achint Thomas, Yashar Mehdad, and Yi Chang. Abusive language detection in online user content. In *Proceedings of the 25th international conference on world wide web*, pages 145–153. International World Wide Web Conferences Steering Committee, 2016. 63

- [471] Timo Ojala, Matti Pietikainen, and Topi Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on pattern analysis and machine intelligence*, 24(7):971–987, 2002. 49
- [472] Aude Oliva and Antonio Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International journal of computer vision*, 42(3):145–175, 2001. 192
- [473] Aude Oliva and Antonio Torralba. Building the gist of a scene: The role of global image features in recognition. *Progress in brain research*, 155:23–36, 2006. 46
- [474] Karolina Owczarzak, John M Conroy, Hoa Trang Dang, and Ani Nenkova. An assessment of the accuracy of automatic evaluation in summarization. In *Proceedings of Workshop on Evaluation Metrics and System Comparison for Automatic Summarization*, pages 1–9. Association for Computational Linguistics, 2012. 96
- [475] Ozer Ozdikis, Pinar Senkul, and Halit Oguztuzun. Semantic expansion of tweet contents for enhanced event detection in twitter. In *ASONAM*, pages 20–24, 2012. 106
- [476] Mustafa Ozuysal, Pascal Fua, and Vincent Lepetit. Fast keypoint recognition in ten lines of code, 2007. 51
- [477] François Pachet and Pierre Roy. Analytical features: a knowledge-based approach to audio feature generation. *EURASIP Journal on Audio, Speech, and Music Processing*, 2009:1–23, 2009. 187, 192
- [478] Alok Ranjan Pal and Diganta Saha. An approach to automatic text summarization using wordnet. In *2014 IEEE International Advance Computing Conference (IACC)*, pages 1169–1173. IEEE, 2014. 237
- [479] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning, oct 2010. 157
- [480] George Panagiotopoulos, George Giannakopoulos, and Antonios Liapis. A study on video game review summarization. In *Proceedings of the MultiLing Workshop*, 2019. 90, 93
- [481] Nikolaos Panagiotou, Ioannis Katakis, and Dimitrios Gunopoulos. Detecting events in online social networks: Definitions, trends and challenges. In *Solving Large Scale Learning Tasks. Challenges and Algorithms*, pages 42–84. 2016. 106
- [482] George Papadakis, George Giannakopoulos, and Georgios Paliouras. Graph vs. bag representation models for the topic classification of web documents. *World Wide Web*, 19(5):887–920, 2016. 61, 65, 106, 111
- [483] Georgios Th Papadopoulos, Krishna Chandramouli, Vasileios Mezaris, Ioannis Kompatsiaris, Ebroul Izquierdo, and Michael G Strintzis. A comparative study of classification techniques for knowledge-assisted image analysis. In *2008 Ninth International*

Workshop on Image Analysis for Multimedia Interactive Services, pages 4–7. IEEE, 2008. 189

- [484] Michalis Papakostas, Evangelos Spyrou, Theodoros Giannakopoulos, Giorgos Siantikos, Dimitrios Sgouropoulos, Phivos Mylonas, and Fillia Makedon. Deep visual attributes vs. hand-crafted audio features on multidomain speech emotion recognition. *Computation*, 5(2):26, 2017. 152
- [485] Ji Ho Park and Pascale Fung. One-step and two-step classification for abusive language detection on twitter. *arXiv preprint arXiv:1706.01206*, 2017. 63
- [486] Sun Park, Ju-Hong Lee, Chan-Min Ahn, Jun Sik Hong, and Seok-Ju Chun. Query based summarization using non-negative matrix factorization. In Bogdan Gabrys, Robert J. Howlett, and Lakhmi C. Jain, editors, *KES (3)*, volume 4253 of *Lecture Notes in Computer Science*, pages 84–89. Springer, 2006. 122
- [487] Robert Parker, David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. English gigaword. *Linguistic Data Consortium*, 2011. 200
- [488] Janne Parkkila, Filip Radulovic, Daniel Garijo, María Poveda-Villalón, Jouni Ikonen, Jari Porras, and Asuncion Gomez-Perez. An ontology for videogame interoperability. *Multimedia Tools and Applications*, 76, 2016. 101
- [489] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011. 250
- [490] Geoffroy Peeters, Bruno L Giordano, Patrick Susini, Nicolas Misdariis, and Stephen McAdams. The timbre toolbox: Extracting audio descriptors from musical signals. *The Journal of the Acoustical Society of America*, 130(5):2902–2916, 2011. 49
- [491] Soo-Chang Pei and Chao-Nan Lin. Image normalization for pattern recognition. *Image and Vision computing*, 13(10):711–723, 1995. 38
- [492] Hanchuan Peng, Fuhui Long, and Chris Ding. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on pattern analysis and machine intelligence*, 27(8):1226–1238, 2005. 52
- [493] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014. 65, 190, 191, 229, 230, 235, 249
- [494] Florent Perronnin and Christopher Dance. Fisher kernels on visual vocabularies for image categorization. In *2007 IEEE conference on computer vision and pattern recognition*, pages 1–8. IEEE, 2007. 47, 51

- [495] Isabella Peters and Wolfgang G Stock. Folksonomy and information retrieval. *Proceedings of the American Society for Information Science and Technology*, 44(1):1–28, 2007. 213
- [496] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018. 191
- [497] Matthew E Peters, Mark Neumann, Robert L Logan IV, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A Smith. Knowledge enhanced contextual word representations. *arXiv preprint arXiv:1909.04164*, 2019. 187, 193
- [498] Matti Pietikäinen, Topi Mäenpää, and Jaakko Viertola. Color texture classification with color histograms and local binary patterns. In *Workshop on Texture Analysis in Machine Vision*, pages 109–112. Citeseer, 2002. 47, 49
- [499] Aggelos Pikrakis, Theodoros Giannakopoulos, and Sergios Theodoridis. A dynamic programming approach to speech/music discrimination of radio recordings. In *Signal Processing Conference, 2007 15th European*, pages 1226–1230. IEEE, 2007. 152
- [500] Aggelos Pikrakis, Theodoros Giannakopoulos, and Sergios Theodoridis. A speech/music discriminator of radio recordings based on dynamic programming and bayesian networks. *IEEE Transactions on Multimedia*, 10(5):846–857, 2008. 152
- [501] Aggelos Pikrakis and Sergios Theodoridis. Speech-music discrimination: A deep learning perspective. In *2014 22nd European Signal Processing Conference (EUSIPCO)*, pages 616–620. IEEE, 2014. 152
- [502] Mohammad Taher Pilehvar, Jose Camacho-Collados, Roberto Navigli, and Nigel Collier. Towards a seamless integration of word senses into downstream nlp applications. *arXiv preprint arXiv:1710.06632*, 2017. 203, 232
- [503] N. Pittaras, G. Giannakopoulos, G. Papadakis, and V. Karkaletsis. Text classification with semantically enriched word embeddings. *Natural Language Engineering*, page 1–35, 2020. 187, 188
- [504] Nikiforos Pittaras, George Giannakopoulos, Leonidas Tsekouras, and Iraklis Varlamis. Document clustering as a record linkage problem. In *Proceedings of the ACM Symposium on Document Engineering 2018*, page 39. ACM, 2018. 135, 138, 141
- [505] Nikiforos Pittaras, Foteini Markatopoulou, Vasileios Mezaris, and Ioannis Patras. Comparison of fine-tuning and extension strategies for deep convolutional neural networks. In *International Conference on Multimedia Modeling*, pages 102–114. Springer, 2017. 53
- [506] Konstantinos N Plataniotis and Anastasios N Venetsanopoulos. *Color image processing and applications*. Springer Science & Business Media, 2013. 48

- [507] Elvys Linhares Pontes, Juan-Manuel Torres-Moreno, and Andréa Carneiro Linhares. Lia-rag: a system based on graphs and divergence of probabilities applied to speech-to-text summarization. *CoRR*, abs/1601.07124, 2016. 124
- [508] MF Porter. An algorithm for suffix stripping. *Program: electronic library & information systems*, 40(3):211–218, 2006. 87, 203
- [509] Michael Portnoff. Time-frequency representation of digital signals and systems based on short-time fourier analysis. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(1):55–69, 1980. 156
- [510] Matt Post and Shane Bergsma. Explicit and implicit syntactic features for text classification. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 866–872, 2013. 235
- [511] Ligaj Pradhan, Chengcui Zhang, and Steven Bethard. Towards extracting coherent user concerns and their hierarchical organization from user reviews. In *2016 IEEE 17th International Conference on Information Reuse and Integration (IRI)*, pages 582–590. IEEE, 2016. 103
- [512] Dragomir Radev, Jahna Otterbacher, Adam Winkel, and Sasha B. Goldensohn. NewsInEssence: summarizing online news topics. *Commun. ACM*, 48(10):95–98, oct 2005. 119, 121
- [513] Dragomir R. Radev, Hongyan Jing, and Małgorzata Budzikowska. Centroid-based summarization of multiple documents: sentence extraction utility-based evaluation, and user studies. *CoRR*, cs.CL/0005020, 2000. 122, 127, 235
- [514] Richard J. Radke, Srinivas Andra, Omar Al-Kofahi, and Badrinath Roysam. Image change detection algorithms: a systematic survey. *IEEE Trans. Image Process.*, 14(3):294–307, 2005. 103
- [515] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*, 2019. 53
- [516] Yves Raimond, Samer A Abdallah, Mark B Sandler, and Frederick Giasson. The music ontology. In *ISMIR*, volume 2007, page 8th. Citeseer, 2007. 40, 181, 183
- [517] William M Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, 66(336):846–850, 1971. 39
- [518] C.R. Rao and V.N. Gudivada. *Computational Analysis and Understanding of Natural Languages: Principles, Methods and Applications*. Handbook of Statistics. Elsevier Science, 2018. 74
- [519] Adwait Ratnaparkhi. A maximum entropy model for part-of-speech tagging. In *Conf. Empir. Methods Nat. Lang. Process.*, 1996. 38

- [520] Mahdyar Ravanbakhsh, Hossein Mousavi, Mohammad Rastegari, Vittorio Murino, and Larry S Davis. Action recognition with image based cnn features. *arXiv preprint arXiv:1512.03980*, 2015. 173
- [521] Muhammad Imran Razzak, Saeeda Naz, and Ahmad Zaib. Deep learning for medical image processing: Overview, challenges and the future. *Classification in BioApps*, pages 323–350, 2018. 37
- [522] Radim Rehurek, Petr Sojka, et al. Gensim—statistical semantics in python. 2011. 238
- [523] Douglas Reynolds. Gaussian mixture models, 2009. 49
- [524] Rafael Ribaldo, Ademar Takeo Akabane, Lucia Helena Machado Rino, Thiago Alex Pardo, and re Salgueiro. Graph-based methods for multi-document summarization: Exploring relationship maps, complex networks and discourse information. In Helena de Medeiros Caseli, Aline Villavicencio, António J. S. Teixeira, and Fernando Perdigão, editors, *PROPOR*, volume 7243 of *Lecture Notes in Computer Science*, pages 260–271. Springer, 2012. 124
- [525] Irina Rish. An empirical study of the naive bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, volume 3, pages 41–46, 2001. 50
- [526] Matthew Roach, John Mason, and Li-Qun Xu. Video genre verification using both acoustic and visual modes. In *Multimedia Signal Processing, 2002 IEEE Workshop on*, pages 157–160. IEEE, 2002. 151, 153, 154
- [527] Matthew Roach and John S Mason. Classification of video genre using audio. In *Seventh European Conference on Speech Communication and Technology*, 2001. 151
- [528] Martín Rocamora and Perfecto Herrera. Comparing audio descriptors for singing voice detection in music audio files. In *Brazilian symposium on computer music, 11th. san pablo, brazil*, volume 26, page 27, 2007. 49
- [529] M Rodriguez, J Hidalgo, and B Agudo. Using wordnet to complement training information in text categorization. In *Proceedings of 2nd International Conference on Recent Advances in Natural Language Processing II: Selected Papers from RANLP*, volume 97, pages 353–364, 2000. 236
- [530] Lior Rokach and Oded Maimon. Clustering methods. In *Data mining and knowledge discovery handbook*, pages 321–352. Springer, 2005. 43
- [531] Gaetano Rossiello, Pierpaolo Basile, and Giovanni Semeraro. Centroid-based text summarization through compositionality of word embeddings. In George Giannakopoulos, Elena Lloret, John M. Conroy, Josef Steinberger, Marina Litvak, Peter A. Rankel, and Benoît Favre, editors, *MultiLing@EACL*, pages 12–21. Association for Computational Linguistics, 2017. 125

- [532] Sascha Rothe and Hinrich Schütze. Autoextend: Extending word embeddings to embeddings for synsets and lexemes. *arXiv preprint arXiv:1507.01127*, 2015. 202
- [533] Sebastian Ruder, Ivan Vulić, and Anders Søgaard. A survey of cross-lingual word embedding models. *Journal of Artificial Intelligence Research*, 65:569–631, 2019. 201
- [534] David E Rumelhart. Learning internal representations by back-propagating errors. *Parallel distributed processing: Explorations in the microstructure of cognition*, 1:318–362, 1986. 50
- [535] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, and Michael Bernstein. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. 55, 157
- [536] James A Russell. A circumplex model of affect. *Journal of personality and social psychology*, 39(6):1161, 1980. 181, 183
- [537] Stuart Russell, Peter Norvig, and Artificial Intelligence. A modern approach. *Artificial Intelligence. Prentice-Hall, Englewood Cliffs*, 25(27):79–80, 1995. 66
- [538] Owen Sacco, Antonios Liapis, and Georgios N. Yannakakis. Game character ontology (gco): A vocabulary for extracting and describing game character information from web content. In *Proceedings of the International Conference on Semantic Systems*, 2017. 101
- [539] Horacio Saggion. Multilingual multidocument summarization tools and evaluation. In Nicoletta Calzolari, Khalid Choukri, Aldo Gangemi, Bente Maegaard, Joseph Mariani, Jan Odijk, and Daniel Tapia, editors, *LREC*, pages 1312–1317. European Language Resources Association (ELRA), 2006. 119
- [540] Hassan Saif, Miriam Fernandez, and Harith Alani. Contextual semantics for sentiment analysis of twitter. *Information Processing & Management*, 52, 2015. 88
- [541] Haçsim Sak, Andrew Senior, and Françoise Beaufays. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In *Fifteenth annual conference of the international speech communication association*, 2014. 152
- [542] Haçsim Sak, Andrew Senior, Kanishka Rao, and Françoise Beaufays. Fast and accurate recurrent neural network acoustic models for speech recognition. *arXiv preprint arXiv:1507.06947*, 2015. 152
- [543] Haji Mohammad Saleem, Kelly P Dillon, Susan Benesch, and Derek Ruths. A web of hate: Tackling hateful speech in online social spaces. *arXiv preprint arXiv:1709.10159*, 2017. 63, 64
- [544] Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523, 1988. 48, 74, 153, 188, 197, 200, 210, 234, 239

- [545] Gerard Salton, Anita Wong, and Chung-Shu Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975. 38, 48, 74, 184, 234
- [546] Gerard Salton and C.S. Yang. On the specification of term values in automatic indexing. *Journal of Documentation*, 29(4):351–372, 1973. 87
- [547] Suhas G. Salve and Kal Jondhale. Shape matching and object recognition using shape contexts, 2010. 49
- [548] David Sánchez, Montserrat Batet, David Isern, and Aida Valls. Ontology-based semantic similarity: A new feature-based approach. *Expert Systems with Applications*, 39, 2012. 101
- [549] Mark Sanderson and W. Bruce Croft. Deriving concept hierarchies from text. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, 1999. 103
- [550] Jagan Sankaranarayanan, Hanan Samet, Benjamin E Teitler, Michael D Lieberman, and Jon Sperling. Twitterstand: news in tweets. In *SIGSPATIAL*, pages 42–51, 2009. 106
- [551] Simone Scardapane, Danilo Comminiello, Michele Scarpiniti, and Aurelio Uncini. Music classification using extreme learning machines. In *2013 8th International Symposium on Image and Signal Processing and Analysis (ISPA)*, pages 377–381. IEEE, 2013. 152
- [552] Thomas Schandler and Andreas Blumauer. Poolparty: SKOS thesaurus management utilizing linked data. In *ESWC*, pages 421–425, 2010. 113
- [553] Jan Schlüter and Sebastian Böck. Musical onset detection with convolutional neural networks. In *6th International Workshop on Machine Learning and Music (MML), Prague, Czech Republic*, 2013. 152
- [554] Jan Schlüter and Sebastian Böck. Improved musical onset detection with convolutional neural networks. In *2014 ieee international conference on acoustics, speech and signal processing (icassp)*, pages 6979–6983. IEEE, 2014. 152
- [555] Helmut Schmid. Treetagger-a language independent part-of-speech tagger. <http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/>, 1994. 188
- [556] Jürgen Schmidhuber. Deep learning in neural networks: An overview, jan 2015. 149
- [557] Anna Schmidt and Michael Wiegand. A survey on hate speech detection using natural language processing. In *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media*, pages 1–10, 2017. 64

- [558] Nathan Schneider and Chuck Wooters. The nltk framenet api: Designing for discoverability with a rich linguistic resource. *arXiv preprint arXiv:1703.07438*, 2017. 181
- [559] Marc Schroder, Elisabetta Bevacqua, Roddy Cowie, Florian Eyben, Hatice Gunes, Dirk Heylen, Mark Ter Maat, Gary McKeown, Sathish Pammi, and Maja Pantic. Building autonomous sensitive artificial listeners. *IEEE Transactions on Affective Computing*, 3(2):165–183, 2012. 154
- [560] Christian Schudt, Ivan Laptev, and Barbara Caputo. Recognizing human actions: a local svm approach. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 3, pages 32–36. IEEE, 2004. 150, 164, 165
- [561] Andreas Schwil. Fundamental ideas of computer science. *Bulletin-European Association for Theoretical Computer Science*, 53:274–274, 1994. 39
- [562] Sam Scott and Stan Matwin. Text classification using wordnet hypernyms. *Usage of WordNet in Natural Language Processing Systems*, 1998. 200, 203, 236
- [563] Sam Scott and Stan Matwin. Feature engineering for text classification. *ICML*, 99:379–388, 1999. 47, 48, 235
- [564] Federico Scozzafava, Alessandro Raganato, Andrea Moro, and Roberto Navigli. Automatic identification and disambiguation of concepts and named entities in the multilingual wikipedia. In *Congress of the Italian Association for Artificial Intelligence*, pages 357–366. Springer, 2015. 202
- [565] Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, 34(1):1–47, 2002. 43, 48
- [566] Ervin Sejdić, Igor Djurović, and Jin Jiang. Time–frequency feature representation using energy concentration: An overview of recent advances. *Digital signal processing*, 19(1):153–183, 2009. 56
- [567] Ali Selamat, Hidekazu Yanagimoto, and Sigeru Omatsu. Web news classification using neural networks based on pca. In *SICE-ANNUAL CONFERENCE-*, number 4, pages 2389–2394. SICE; 1999, 2002. 47, 51
- [568] Piotr Semberecki and Henryk Maciejewski. Distributed classification of text documents on apache spark platform. In *International Conference on Artificial Intelligence and Soft Computing*, pages 621–630. Springer, 2016. 136
- [569] Ishwar K Sethi, Ioana L Coman, and Daniela Stan. Mining association rules between low-level image features and high-level concepts. In *Data Mining and Knowledge Discovery: Theory, Tools, and Technology III*, volume 4384, pages 279–290. International Society for Optics and Photonics, 2001. 46, 191
- [570] Jianbo Shi and Carlo Tomasi. Good features to track. Technical report, Cornell University, 1993. 48

- [571] Weijia Shi, Muhan Chen, Pei Zhou, and Kai-Wei Chang. Retrofitting contextualized word embeddings with paraphrases. *arXiv preprint arXiv:1909.09700*, 2019. 187, 191
- [572] Connor Shorten and Taghi M. Khoshgoftaar. A survey on image data augmentation for deep learning. 6, 2019. 38
- [573] Panagiotis Sidiropoulos, Vasileios Mezaris, and Ioannis Kompatsiaris. Enhancing video concept detection with the use of tomographs. In *Image Processing (ICIP), 2013 20th IEEE International Conference on*, pages 3991–3995. IEEE, 2013. 152
- [574] Leslie F. Sikos. The semantic gap, 2017. 180
- [575] Carlos N Silla and Alex A Freitas. A survey of hierarchical classification across different application domains. *Data Mining and Knowledge Discovery*, 22(1):31–72, 2011. 184
- [576] Catarina Silva and Bernardete Ribeiro. The importance of stop word removal on recall values in text categorization. In *Neural Networks, 2003. Proceedings of the International Joint Conference on*, volume 3, pages 1661–1666. IEEE, 2003. 38
- [577] Leandro Araújo Silva, Mainack Mondal, Denzil Correa, Fabricio Benevenuto, and Ingmar Weber. Analyzing the targets of hate in online social media. In *ICWSM*, pages 687–690, 2016. 63
- [578] Mattia Silvestri, Michele Lombardi, and Michela Milano. Injecting domain knowledge in neural networks: a controlled experiment on a constrained problem. *arXiv preprint arXiv:2002.10742*, 2020. 180
- [579] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, pages 568–576, 2014. 154, 155
- [580] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 47, 55, 56, 155
- [581] Sukhpreet Singh. Optical character recognition techniques: a survey. *Journal of Emerging Trends in Computing and Information Sciences*, 4(6):545–550, 2013. 37
- [582] Blaž Škrlj, Matej Martinc, Jan Kralj, Nada Lavrač, and Senja Pollak. tax2vec: Constructing interpretable features from taxonomies for short text classification. *Computer Speech & Language*, page 101104, 2020. 187, 188
- [583] Malcolm Slaney. Auditory toolbox. *Interval Research Corporation, Tech. Rep*, 10(1998), 1998. 49
- [584] Alan F Smeaton, Paul Over, and Wessel Kraaij. Trecvid: Evaluating the effectiveness of information retrieval tasks on digital video. In *Proceedings of the 12th annual ACM international conference on Multimedia*, pages 652–655. ACM, 2004. 150

- [585] Cees GM Snoek and Marcel Worring. Multimodal video indexing: A review of the state-of-the-art. *Multimedia tools and applications*, 25(1):5–35, 2005. 149
- [586] Cees GM Snoek, Marcel Worring, and Arnold WM Smeulders. Early versus late fusion in semantic video analysis. In *Proceedings of the 13th annual ACM international conference on Multimedia*, pages 399–402. ACM, 2005. 154
- [587] Richard Socher, Eric H Huang, Jeffrey Pennin, Christopher D Manning, and Andrew Y Ng. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In *Advances in neural information processing systems*, pages 801–809, 2011. 54
- [588] Marina Sokolova and Guy Lapalme. A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4):427 – 437, 2009. 39
- [589] Sheetal S Sonawane and Parag A Kulkarni. Graph based representation and analysis of text document: A survey of techniques. *International Journal of Computer Applications*, 96(19), 2014. 38
- [590] Min Song, Won Chul Kim, Dahee Lee, Go Eun Heo, and Keun Young Kang. PKDE4J: Entity and relation extraction for public knowledge discovery. *Journal of Biomedical Informatics*, 57:320–332, 2015. 103
- [591] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 154, 164, 165
- [592] Karen Spärck Jones. Automatic summarising: Factors and directions. In I. Mani and M. Maybury, editors, *Advances in Automatic Text Summarisation*. MIT Press, Cambridge MA, 1998. Also in asXiv:cmp-lg/9805011. 120
- [593] Munirathnam Srikanth, Joshua Varner, Mitchell Bowden, and Dan I. Moldovan. Exploiting ontologies for automatic image annotation. In Ricardo A. Baeza-Yates, Nivio Ziviani, Gary Marchionini, Alistair Moffat, and John Tait, editors, *SIGIR*, pages 552–558. ACM, 2005. 187, 191
- [594] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014. 55
- [595] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhudinov. Unsupervised learning of video representations using lstms. In *International conference on machine learning*, pages 843–852, 2015. 153
- [596] Ryan Stables, Sean Enderby, Brecht De Man, György Fazekas, and Joshua D Reiss. Safe: A system for extraction and retrieval of semantic audio descriptors. 2014. 49

- [597] Mark Steyvers and Tom Griffiths. Probabilistic Topic Models. page 15, 2017. 73, 85
- [598] Dmitry Storcheus, Afshin Rostamizadeh, and Sanjiv Kumar. A survey of modern questions and challenges in feature extraction. In *Feature Extraction: Modern Questions and Challenges*, pages 1–18. PMLR, 2015. 39
- [599] Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and policy considerations for deep learning in nlp. *arXiv preprint arXiv:1906.02243*, 2019. 39, 180
- [600] Bob L Sturm. A survey of evaluation in music genre recognition. In *International Workshop on Adaptive Multimedia Retrieval*, pages 29–66. Springer, 2012. 37
- [601] Pero Subasic and Alison Huettner. Affect analysis of text using fuzzy semantic typing. *IEEE Transactions on Fuzzy Systems*, 2:483–496, 2001. 88
- [602] Thamarai Subramaniam, Hamid A Jalab, and Alaa Y Taqa. Overview of textual anti-spam filtering techniques. *International Journal of Physical Sciences*, 5(12):1869–1882, 2010. 37
- [603] Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, pages 697–706, 2007. 181, 182
- [604] Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. How to fine-tune bert for text classification? In *China National Conference on Chinese Computational Linguistics*, pages 194–206. Springer, 2019. 47, 55
- [605] Shiliang Sun. A survey of multi-view machine learning. *Neural Computing and Applications*, 23(7-8):2031–2038, 2013. 154
- [606] Yiwei Sun and Shabnam Ghaffarzadegan. An ontology-aware framework for audio event classification. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 321–325. IEEE, 2020. 187, 194
- [607] Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. Ernie: Enhanced representation through knowledge integration. *arXiv preprint arXiv:1904.09223*, 2019. 187, 193
- [608] Shan Suthaharan. Machine learning models and algorithms for big data classification. *Integr. Ser. Inf. Syst.*, 36:1–12, 2016. 37
- [609] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-first AAAI conference on artificial intelligence*, 2017. 55
- [610] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. 47, 55, 56, 176

- [611] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016. 55, 56
- [612] Kai Sheng Tai, Richard Socher, and Christopher D Manning. Improved semantic representations from tree-structured long short-term memory networks. *arXiv preprint arXiv:1503.00075*, 2015. 191
- [613] Kevin Tang, Li Fei-Fei, and Daphne Koller. Learning latent temporal structure for complex event detection. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1250–1257. IEEE, 2012. 153
- [614] Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. Efficient transformers: A survey. *arXiv preprint arXiv:2009.06732*, 2020. 43
- [615] Simone Teufel and Marc Moens. Summarizing scientific articles: Experiments with relevance and rhetorical status. *Computational Linguistics*, 28(4):409–445, 2002. 119, 123
- [616] Mike Thelwall, Kevan Buckley, and Georgios Paltoglou. Sentiment strength detection for the social web. *Journal of the American Society for Information Science and Technology*, 63(1):163–173, 2012. 88, 101
- [617] Theodoros Theodorou, Iosif Mporas, and Nikos Fakotakis. An overview of automatic audio segmentation. *International Journal of Information Technology and Computer Science (IJITCS)*, 6(11):1, 2014. 38
- [618] Stefan Thomas, Christian Beutnmüller, Xose de la Puente, Robert Remus, and Stefan Bordag. ExB Text Summarizer. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 260–269, Prague, Czech Republic, 2015. Association for Computational Linguistics. 75, 125
- [619] Rafael Torralbo, Enrique Alfonsena, Antonio Moreno-Sandoval, and José María Guirao. Automatic generation of term definitions using multidocument summarisation from the web. *CROSSING BARRIERS IN TEXT SUMMARIZATION RESEARCH*, page 32, 2005. 122
- [620] Bruno Trstenjak, Sasa Mikac, and Dzenana Donko. Knn with tf-idf based framework for text categorization. 69:1356–1364, 2014. 47, 48
- [621] Ba Tu Truong and Chitra Dorai. Automatic genre identification for content-based video categorization. In *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, volume 4, pages 230–233. IEEE, 2000. 153, 154
- [622] Leonidas Tsekouras, Iraklis Varlamis, and George Giannakopoulos. A graph-based text similarity measure that employs named entity information. In *Proc. Int. Conf. Recent Adv. Nat. Lang. Process. RANLP 2017*, pages 765–771, 2017. 72, 135

- [623] Marco Turchi, Josef Steinberger, Mijail A. Kabadjov, and Ralf Steinberger. Using parallel corpora for multilingual (multi-document) summarisation evaluation. In Maristella Agosti, Nicola Ferro, Carol Peters, Maarten de Rijke, and Alan F. Smeaton, editors, *CLEF*, volume 6360 of *Lecture Notes in Computer Science*, pages 52–63. Springer, 2010. 119, 121
- [624] Joseph Turian, Lev Ratinov, and Yoshua Bengio. Word representations: A simple and general method for semi-supervised learning. pages 384–394, 2010. 54
- [625] P. D. Turney and P. Pantel. From Frequency to Meaning: Vector Space Models of Semantics. *Journal of Artificial Intelligence Research*, 37:141–188, February 2010. 74
- [626] Peter D. Turney. Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, 2002. 87, 88
- [627] Peter D Turney and Patrick Pantel. From frequency to meaning : Vector space models of semantics. 37:141–188, 2010. 184
- [628] Tinne Tuytelaars. Dense interest points. In *The Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010, San Francisco, CA, USA, 13-18 June 2010*, pages 2281–2288. IEEE Computer Society, 2010. 48
- [629] Tinne Tuytelaars and Krystian Mikolajczyk. Local invariant feature detectors: a survey. *Foundations and trends® in computer graphics and vision*, 3(3):177–280, 2008. 48
- [630] George Tzanetakis. Audio feature extraction. *Music data mining*, pages 49–74, 2011. 49
- [631] Yusuke Uchida. Local feature detectors, descriptors, and image representations: a survey. *arXiv preprint arXiv:1607.08368*, 2016. 43
- [632] Karen Ullrich, Jan Schlüter, and Thomas Grill. Boundary detection in music structure analysis using convolutional neural networks. In *ISMIR*, pages 417–422, 2014. 152
- [633] JW Uys, ND Du Preez, and EW Uys. Leveraging unstructured information using topic modelling. In *PICMET'08-2008 Portland International Conference on Management of Engineering & Technology*, pages 955–961. IEEE, 2008. 43
- [634] Stijn Marinus Van Dongen. *Graph clustering by flow simulation*. PhD thesis, 2000. 129, 133
- [635] David A Van Dyk and Xiao-Li Meng. The art of data augmentation. *Journal of Computational and Graphical Statistics*, 10(1):1–50, 2001. 38
- [636] Vladimir Vapnik, Isabel Guyon, and Trevor Hastie. Support vector machines. *Mach. Learn*, 20(3):273–297, 1995. 48

- [637] Vladimir Vapnik and Akshay Vashist. A new learning paradigm: Learning using privileged information. *Neural networks*, 22(5-6):544–557, 2009. 191
- [638] Ramakrishna Varadarajan and Vagelis Hristidis. A system for query-specific document summarization. In Philip S. Yu, Vassilis J. Tsotras, Edward A. Fox, and Bing Liu, editors, *CIKM*, pages 622–631. ACM, 2006. 122, 123
- [639] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 92
- [640] Marta Vicente, Oscar Alcón, and Elena Lloret. The university of alicante at multiling 2015: approach, results and further insights. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 250–259, 2015. 75, 125, 236
- [641] S Vijayarani, Ms J Ilamathi, and Ms Nithya. Preprocessing techniques for text mining—an overview. *International Journal of Computer Science & Communication Networks*, 5(1):7–16, 2015. 38
- [642] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103, 2008. 193
- [643] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164, 2015. 161
- [644] Julia Vogel and Bernt Schiele. Semantic modeling of natural scenes for content-based image retrieval. *International Journal of Computer Vision*, 72(2):133–157, 2007. 187, 189
- [645] Denny Vrandečić and Markus Krötzsch. Wikidata: a free collaborative knowledge-base. *Communications of the ACM*, 57(10):78–85, 2014. 181, 182
- [646] Ivan Vulic and Nikola Mrkšić. Specialising word vectors for lexical entailment. In *Proceedings of NAACL-HLT*, pages 1134–1145. The Association for Computational Linguistics, 2018. 201
- [647] Li L. Huang T. Gao Z. Mao L. & Huang F. Wan, S. Cist system report for sigdial multiling 2015. In *SIGDIAL*, 2015. 125, 126
- [648] Xiaojun Wan, Houping Jia, Shanshan Huang, and Jianguo Xiao. Summarizing the differences in multilingual news. In Wei-Ying Ma, Jian-Yun Nie, Ricardo Baeza-Yates, Tat-Seng Chua, and W. Bruce Croft, editors, *SIGIR*, pages 735–744. ACM, 2011. 119

- [649] Heng Wang, Alexander Kläser, Cordelia Schmid, and Cheng-Lin Liu. Dense trajectories and motion boundary descriptors for action recognition. *International journal of computer vision*, 103(1):60–79, 2013. 153
- [650] Heng Wang and Cordelia Schmid. Action recognition with improved trajectories. In *Proceedings of the IEEE international conference on computer vision*, pages 3551–3558, 2013. 153, 154
- [651] Jia-Ching Wang, Jhing-Fa Wang, Kuok Wai He, and Cheng-Shu Hsu. Environmental sound classification using hybrid svm/knn classifier and mpeg-7 audio low-level descriptor. In *The 2006 IEEE international joint conference on neural network proceedings*, pages 1731–1735. IEEE, 2006. 49
- [652] Ju-Chiang Wang, Hung-Yi Lo, Shyh-Kang Jeng, and Hsin-Min Wang. Mirex 2010: Audio classification using semantic transformation and classifier ensemble. In *Proc. of The 6th International WOCMAT & New Media Conference (WOCMAT 2010)*, pages 2–5. Citeseer, 2010. 190
- [653] Limin Wang, Yu Qiao, and Xiaou Tang. Action recognition with trajectory-pooled deep-convolutional descriptors. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4305–4314, 2015. 153
- [654] Peng Wang, Rui Cai, and Shi-Qiang Yang. A hybrid approach to news video classification multimodal features. In *Information, communications and signal processing, 2003 and fourth pacific rim conference on multimedia. Proceedings of the 2003 joint conference of the fourth international conference on*, volume 2, pages 787–791. IEEE, 2003. 154
- [655] Yuxuan Wang, Yutai Hou, Wanxiang Che, and Ting Liu. From static to dynamic word representations: a survey. *International Journal of Machine Learning and Cybernetics*, pages 1–20, 2020. 43
- [656] Zhiyong Wang, Bin Lu, Zheru Chi, and David Dagan Feng. Leaf image classification with shape context and SIFT descriptors. In Andrew P. Bradley and Paul T. Jackway, editors, *2011 International Conference on Digital Image Computing: Techniques and Applications (DICTA), Noosa, QLD, Australia, December 6-8, 2011*, pages 650–654. IEEE Computer Society, 2011. 47, 49
- [657] William Warner and Julia Hirschberg. Detecting hate speech on the world wide web. In *Proceedings of the Second Workshop on Language in Social Media*, pages 19–26. Association for Computational Linguistics, 2012. 63, 64
- [658] Amy Beth Warriner, Victor Kuperman, and Marc Brysbaert. Norms of valence, arousal, and dominance for 13,915 english lemmas. *Behavior research methods*, 45(4):1191–1207, 2013. 181, 183, 191

- [659] Zeerak Waseem. Are you a racist or am i seeing things? annotator influence on hate speech detection on twitter. In *Proceedings of the first workshop on NLP and computational social science*, pages 138–142, 2016. 63
- [660] Zeerak Waseem and Dirk Hovy. Hateful symbols or hateful people? predictive features for hate speech detection on twitter. In *Proceedings of the NAACL student research workshop*, pages 88–93, 2016. 64, 67
- [661] Colin Whittaker, Brian Ryner, and Marria Nazif. Large-scale automatic classification of phishing pages. 2010. 37
- [662] Janyce Wiebe. Learning subjective adjectives from corpora. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, page 735–740, 2000. 88
- [663] E. Wold, T. Blum, D. Keislar, and J. Wheaten. Content-based classification, search, and retrieval of audio. 3:27–36, 1996. 47, 49
- [664] Martin Wöllmer, Moritz Kaiser, Florian Eyben, Björn Schuller, and Gerhard Rigoll. Lstm-modeling of continuous emotions in an audiovisual affect recognition framework. *Image and Vision Computing*, 31(2):153–163, 2013. 154
- [665] Raymond E Wright. Logistic regression. 1995. 250
- [666] Chia-Wei Wu and Chao-Lin Liu. Ontology-based text summarization for business news articles. In Narayan C. Debnath, editor, *Computers and Their Applications*, pages 389–392. ISCA, 2003. 119
- [667] Lei Wu, Steven CH Hoi, and Nenghai Yu. Semantics-preserving bag-of-words models and applications. *IEEE Transactions on Image Processing*, 19(7):1908–1920, 2010. 187, 192
- [668] Zuxuan Wu, Yu-Gang Jiang, Xi Wang, Hao Ye, and Xiangyang Xue. Multi-stream multi-class fusion of deep networks for video classification. In *Proceedings of the 2016 ACM on Multimedia Conference*, pages 791–800. ACM, 2016. 155, 173
- [669] Zuxuan Wu, Yu-Gang Jiang, Xi Wang, Hao Ye, Xiangyang Xue, and Jun Wang. Fusing multi-stream deep networks for video classification. *arXiv preprint arXiv:1509.06086*, 2015. 173
- [670] Zuxuan Wu, Ting Yao, Yanwei Fu, and Yu-Gang Jiang. Deep learning for video classification and captioning. *arXiv preprint arXiv:1609.06782*, 2016. 150, 153
- [671] Guang Xiang, Bin Fan, Ling Wang, Jason Hong, and Carolyn Rose. Detecting offensive tweets via topical feature discovery over a large scale twitter corpus. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 1980–1984. ACM, 2012. 63, 64

- [672] Chang Xu, Yalong Bai, Jiang Bian, Bin Gao, Gang Wang, Xiaoguang Liu, and Tie-Yan Liu. Rc-net: A general framework for incorporating knowledge into word representations. In *Proceedings of the 23rd ACM international conference on conference on information and knowledge management*, pages 1219–1228. ACM, 2014. 201
- [673] Chang Xu, Dacheng Tao, and Chao Xu. A survey on multi-view learning. *arXiv preprint arXiv:1304.5634*, 2013. 154
- [674] Min Xu, L-T Chia, and Jesse Jin. Affective content analysis in comedy and horror videos by audio emotional event detection. In *Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on*, pages 4–pp. IEEE, 2005. 151
- [675] Ying Xu, Jey Han Lau, Timothy Baldwin, and Trevor Cohn. Decoupling encoder and decoder networks for abstractive document summarization. In *Proceedings of the MultiLing 2017 Workshop on Summarization and Summary Evaluation Across Source Types and Genres*, pages 7–11, 2017. 125
- [676] Zhi Xu and Sencun Zhu. Filtering offensive language in online communities using grammatical relations. In *Proceedings of the Seventh Annual Collaboration, Electronic Messaging, Anti-Abuse and Spam Conference*, pages 1–10, 2010. 62, 63
- [677] Gui-Rong Xue, Wenyuan Dai, Qiang Yang, and Yong Yu. Topic-bridged plsa for cross-domain text classification. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 627–634. ACM, 2008. 47, 51
- [678] Nestor Yague-Martinez, Francesco De Zan, and Pau Prats-Iraola. Coregistration of interferometric stacks of sentinel-1 TOPS data. *IEEE Geosci. Remote Sensing*, 14(7):1002–1006, 2017. 109
- [679] Junji Yamato, Jun Ohya, and Kenichiro Ishii. Recognizing human action in time-sequential images using hidden markov model. In *Computer Vision and Pattern Recognition, 1992. Proceedings CVPR’92., 1992 IEEE Computer Society Conference on*, pages 379–385. IEEE, 1992. 153
- [680] Jianchao Yang, Kai Yu, Yihong Gong, and Thomas S. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, pages 1794–1801. IEEE Computer Society, 2009. 47, 51
- [681] Wankou Yang, Zhenyu Wang, Jun Yin, Changyin Sun, and Karl Ricanek. Image classification using kernel collaborative representation with regularized least square. *Applied Mathematics and Computation*, 222:13–28, 2013. 52
- [682] Xiao Yang, Craig Macdonald, and Iadh Ounis. Using word embeddings in twitter election classification. *Information Retrieval Journal*, 21(2-3):183–207, 2018. 47, 54

- [683] Yi Yang and Shawn Newsam. Comparing sift descriptors and gabor texture features for classification of remote sensed imagery. In *2008 15th IEEE international conference on image processing*, pages 1852–1855. IEEE, 2008. 47, 48, 51
- [684] Yiming Yang. A comparative study on feature selection in text categorization. 1997. 46, 47, 48, 234
- [685] Yiming Yang, Tom Pierce, and Jaime Carbonell. A study of retrospective and on-line event detection. In *SIGIR*, pages 28–36, 1998. 106
- [686] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489, 2016. 188
- [687] Kevin Yauris and Masayu Leylia Khodra. Aspect-based summarization for game review using double propagation. In *Proceedings of the International Conference on Advanced Informatics, Concepts, Theory, and Applications*, 2017. 86, 88, 89
- [688] Hao Ye, Zuxuan Wu, Rui-Wei Zhao, Xi Wang, Yu-Gang Jiang, and Xiangyang Xue. Evaluating two-stream cnn for video classification. In *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*, pages 435–442. ACM, 2015. 154, 160
- [689] Ozge Yeloglu, Evangelos E. Milios, and A. Nur Zincir-Heywood. Multi-document summarization of scientific corpora. In William C. Chu, W. Eric Wong, Mathew J. Palakal, and Chih-Cheng Hung, editors, *SAC*, pages 252–258. ACM, 2011. 119
- [690] Jaya Kumar Yogan, Ong Sing Goh, Basiron Halizah, Hea Choon Ngo, and C Puspala. A review on automatic text summarization approaches. *Journal of Computer Science*, 12(4):178–190, 2016. 37, 234, 235
- [691] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Advances in neural information processing systems*, pages 3320–3328, 2014. 53
- [692] Changsong Yu, Karim Said Barsim, Qiuqiang Kong, and Bin Yang. Multi-level attention model for weakly supervised audio classification. *arXiv preprint arXiv:1803.02353*, 2018. 174
- [693] Liang-Chih Yu, Jin Wang, K Robert Lai, and Xuejie Zhang. Refining word embeddings for sentiment analysis. In *Proceedings of the 2017 conference on empirical methods in natural language processing*, pages 534–539, 2017. 187, 191
- [694] Mo Yu and Mark Dredze. Improving lexical embeddings with semantic knowledge. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 545–550, 2014. 200

- [695] Joe Yue-Hei Ng, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici. Beyond short snippets: Deep networks for video classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4694–4702, 2015. 151, 153
- [696] Bianca Zadrozny. Learning and evaluating classifiers under sample selection bias. In *Proceedings of the twenty-first international conference on Machine learning*, page 114, 2004. 51
- [697] José P. Zagal, Noriko Tomuro, and Andriy Shepitsen. Natural language processing in game studies research: An overview. *Simulation & Gaming*, 43(3):356–373, 2012. 102
- [698] Matei Zaharia, Andrew Chen, Aaron Davidson, Ali Ghodsi, Sue Ann Hong, Andy Konwinski, Siddharth Murching, Tomas Nykodym, Paul Ongilvie, Mani Parkhe, et al. Accelerating the machine learning lifecycle with mlflow. *IEEE Data Eng. Bull.*, 41(4):39–45, 2018. 250
- [699] Matei Zaharia, Reynold S. Xin, Patrick Wendell, Tathagata Das, Michael Armbrust, Ankur Dave, Xiangrui Meng, Josh Rosen, Shivaram Venkataraman, Michael J. Franklin, Ali Ghodsi, Joseph Gonzalez, Scott Shenker, and Ion Stoica. Apache spark: a unified engine for big data processing. *Commun. ACM*, 59(11):56–65, 2016. 135
- [700] Nida M Zaitoun and Musbah J Aqel. Survey on image segmentation techniques. *Procedia Computer Science*, 65:797–806, 2015. 38
- [701] Elias Zavitsanos, Georgios Palioras, George A Vouros, and Sergios Petridis. Learning subsumption hierarchies of ontology concepts from texts. *Web Intelligence and Agent Systems: An International Journal*, 8(1):37–51, 2010. 103
- [702] Matthew D Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012. 176
- [703] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision*, pages 818–833. Springer, 2014. 56, 152
- [704] Sarah Zelikovitz and Haym Hirsh. Using lsi for text classification in the presence of background text. In *Proceedings of the tenth international conference on Information and knowledge management*, pages 113–118. ACM, 2001. 47, 51
- [705] Chiyuan Zhang, Georgios Evangelopoulos, Stephen Voinea, Lorenzo Rosasco, and Tomaso Poggio. A deep representation for invariance and music classification. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6984–6988. IEEE, 2014. 47, 56, 152
- [706] Dehai Zhang, Menglong Cui, Yun Yang, Po Yang, Cheng Xie, Di Liu, Beibei Yu, and Zhibo Chen. Knowledge graph-based image classification refinement. *IEEE Access*, 7:57678–57690, 2019. 187, 194

- [707] Jianguo Zhang, Marcin Marszałek, Svetlana Lazebnik, and Cordelia Schmid. Local features and kernels for classification of texture and object categories: A comprehensive study. *International journal of computer vision*, 73(2):213–238, 2007. 46
- [708] Quan-shi Zhang and Song-Chun Zhu. Visual interpretability for deep learning: a survey. *Frontiers of Information Technology & Electronic Engineering*, 19(1):27–39, 2018. 53
- [709] Si Zhang, Hanghang Tong, Jiejun Xu, and Ross Maciejewski. Graph convolutional networks: a comprehensive review. *Computational Social Networks*, 6(1):1–23, 2019. 194
- [710] Tong Zhang and C-C Jay Kuo. Hierarchical system for content-based audio classification and retrieval. In *Multimedia Storage and Archiving Systems III*, volume 3527, pages 398–409. International Society for Optics and Photonics, 1998. 187, 190
- [711] Tong Zhang and C-C Jay Kuo. Audio content analysis for online audiovisual data segmentation and classification. *IEEE Transactions on speech and audio processing*, 9(4):441–457, 2001. 154
- [712] Wen Zhang, Taketoshi Yoshida, and Xijin Tang. A comparative study of tf* idf, lsi and multi-words for text classification. *Expert Systems with Applications*, 38(3):2758–2765, 2011. 47, 51
- [713] Yi Zhang, Jamie Callan, and Thomas Minka. Novelty and redundancy detection in adaptive filtering. In *SIGIR ’02: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 81–88, New York, NY, USA, 2002. ACM. 127
- [714] Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. Ernie: Enhanced language representation with informative entities. *arXiv preprint arXiv:1905.07129*, 2019. 187, 193
- [715] Lili Zhao and Chunping Li. Ontology based opinion mining for movie reviews. In *Knowledge Science, Engineering and Management*, pages 204–214. Springer Berlin Heidelberg, 2009. 86
- [716] Alice Zheng and Amanda Casari. *Feature engineering for machine learning: principles and techniques for data scientists.* “ O’Reilly Media, Inc.”, 2018. 39
- [717] Fang Zheng, Guoliang Zhang, and Zhanjiang Song. Comparison of different implementations of mfcc. *Journal of Computer science and Technology*, 16(6):582–589, 2001. 151
- [718] Fan Zhu, Ling Shao, Jin Xie, and Yi Fang. From handcrafted to learned representations for human action recognition: A survey. *Image and Vision Computing*, 55:42–52, 2016. 43

- [719] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1):43–76, 2020. 53
- [720] Li Zhuang, Feng Jing, and Xiao-Yan Zhu. Movie review mining and summarization. In *Proceedings of the ACM International Conference on Information and Knowledge Management*, page 43–50, 2006. 86
- [721] Jinyi Zou, Wei Li, Chen Chen, and Qian Du. Scene classification using local and global features with collaborative representation fusion. *Information Sciences*, 348:209–226, 2016. 47, 51
- [722] Zhen Zuo. Sentiment Analysis of Steam Review Datasets using Naive Bayes and Decision Tree Classifier. Technical report, University of Illinois at Urbana–Champaign, 2018. 86, 93