

**Title line 1**

**Title line 2**

**A thesis submitted in partial fulfilment  
of the requirement for the degree of Doctor of Philosophy**

**Name M. Lastname**

**July 2011**

**Cardiff University  
School of Computer Science & Informatics**

**Declaration**

This work has not previously been accepted in substance for any degree and is not concurrently submitted in candidature for any degree.

Signed ..... (candidate)

Date .....

**Statement 1**

This thesis is being submitted in partial fulfillment of the requirements for the degree of PhD.

Signed ..... (candidate)

Date .....

**Statement 2**

This thesis is the result of my own independent work/investigation, except where otherwise stated. Other sources are acknowledged by explicit references.

Signed ..... (candidate)

Date .....

**Statement 3**

I hereby give consent for my thesis, if accepted, to be available for photocopying and for inter-library loan, and for the title and summary to be made available to outside organisations.

Signed ..... (candidate)

Date .....

Copyright © 2011 Name Lastname.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled “GNU Free Documentation License”.

A copy of this document in various transparent and opaque machine-readable formats and related software is available at <http://yourwebsite>.

**To People you care  
for their patience and support.**

# Abstract

We produce interpretable representations, and demonstrate their applicability in interpretable classifiers. Our approach is model-agnostic, given a similarity-based representation, we are able to produce a representation in terms of domain knowledge. We evaluate the interpretability of our representation and provide examples of interpretable classifiers with our representation.

# Acknowledgements

# Contents

|                                  |            |
|----------------------------------|------------|
| <b>Abstract</b>                  | <b>iv</b>  |
| <b>Acknowledgements</b>          | <b>v</b>   |
| <b>Contents</b>                  | <b>vi</b>  |
| <b>List of Publications</b>      | <b>x</b>   |
| <b>List of Figures</b>           | <b>xi</b>  |
| <b>List of Tables</b>            | <b>xii</b> |
| <b>List of Algorithms</b>        | <b>xiv</b> |
| <b>List of Acronyms</b>          | <b>xv</b>  |
| 0.0.1 Definitions . . . . .      | xv         |
| <b>1 Introduction</b>            | <b>1</b>   |
| 1.1 Representations . . . . .    | 2          |
| 1.2 Motivation . . . . .         | 2          |
| 1.2.1 Machine Learning . . . . . | 3          |
| 1.2.2 Directions . . . . .       | 4          |

---

|          |   |          |
|----------|---|----------|
| 1.3      | Interpretability . . . . .                | 5        |
| 1.4      | Thesis Overview / Contributions . . . . . | 6        |
| <b>2</b> | <b>Background</b>                         | <b>7</b> |
| 2.1      | Introduction . . . . .                    | 7        |
| 2.2      | Text Representations . . . . .            | 10       |
| 2.2.1    | Preprocessing . . . . .                   | 10       |
| 2.2.2    | Bag-of-words . . . . .                    | 11       |
| 2.2.3    | Principal Component Analysis . . . . .    | 13       |
| 2.2.4    | Multi-Dimensional Scaling . . . . .       | 13       |
| 2.3      | Classification and Regression . . . . .   | 14       |
| 2.3.1    | Evaluation Metrics . . . . .              | 15       |
| 2.3.2    | Decision Trees . . . . .                  | 16       |
| 2.4      | Neural Networks . . . . .                 | 16       |
| 2.4.1    | Feedforward Neural Networks . . . . .     | 16       |
| 2.4.2    | Word Vectors . . . . .                    | 16       |
| 2.4.3    | Doc2Vec . . . . .                         | 16       |
| 2.5      | Support Vector Machines . . . . .         | 16       |
| 2.6      | Clustering . . . . .                      | 17       |
| 2.6.1    | K-means . . . . .                         | 17       |
| 2.6.2    | Derrac's K-means Variation . . . . .      | 17       |
| 2.7      | Interpretable Representations . . . . .   | 17       |
| 2.7.1    | Topic Models . . . . .                    | 17       |
| 2.8      | Interpretable Representations . . . . .   | 17       |
| 2.8.1    | Word Vectors . . . . .                    | 18       |



---

|          |   |           |
|----------|---|-----------|
| <b>3</b> | <b>Datasets and Semantic Spaces</b>                                   | <b>19</b> |
| 3.1      | Introduction . . . . .  | 19        |
| 3.2      | Datasets . . . . .  | 19        |
| 3.3      | Technical Details . . . . .   | 22        |
| 3.4      | Representations . . . . .   | 23        |
| <b>4</b> | <b>Re-organizing Vector Spaces into Interpretable Representations</b> | <b>25</b> |
| 4.1      | Introduction . . . . .  | 25        |
| 4.2      | Method . . . . .  | 29        |
| 4.2.1    | Obtaining Directions and Rankings From Words . . . . .                | 30        |
| 4.2.2    | Filtering Word-Directions . . . . .                                   | 32        |
| 4.2.3    | Clustering Features . . . . .   | 35        |
| 4.3      | Qualitative Results . . . . .   | 37        |
| 4.3.1    | Datasets . . . . .  | 38        |
| 4.3.2    | Space Types . . . . .   | 38        |
| 4.3.3    | The best-performing directions for each domain . . . . .              | 39        |
| 4.3.4    | Comparing Space Types . . . . .                                       | 41        |
| 4.4      | Quantitative Results . . . . .  | 45        |
| 4.4.1    | Evaluation Method . . . . .   | 45        |
| 4.4.2    | Summary of all Results . . . . .                                      | 50        |
| 4.4.3    | Baseline Representations . . . . .                                    | 52        |
| 4.4.4    | Word Directions . . . . .   | 56        |
| 4.4.5    | Clustered Directions . . . . .  | 58        |
| 4.4.6    | Conclusion . . . . .  | 62        |

---

|          |   |           |
|----------|---|-----------|
| <b>5</b> | <b>Fine-tuning Vector Spaces to Improve Their Directions</b>                | <b>63</b> |
| 5.1      | Introduction . . . . .  | 63        |
| 5.2      | Fine-Tuning Vector Spaces And Their Associated Feature Directions . . . . . | 66        |
| 5.2.1    | Generating Target Rankings . . . . .  | 67        |
| 5.2.2    | Generating Target Feature Values . . . . .                                  | 67        |
| 5.2.3    | Fine-Tuning . . . . .   | 68        |
| 5.3      | Quantitative Evaluation . . . . .   | 69        |
| 5.3.1    | Results . . . . .   | 72        |
| 5.4      | Conclusions . . . . .   | 73        |
| <b>6</b> | <b>Investigating Neural Networks In Terms Of Directions</b>                 | <b>74</b> |
| 6.1      | Chapter 5 . . . . .   | 74        |
| 6.1.1    | Chapter 3 Space Types . . . . .   | 74        |
| <b>7</b> | <b>Appendix</b>   | <b>76</b> |
| 7.1      | Chapter 3 . . . . .   | 76        |
| 7.1.1    | Difference between Representations and Single Directions . . . . .          | 76        |
| 7.1.2    | Class Names and Positive Occurrences . . . . .                              | 78        |
|          | <b>GNU Free Documentation License</b>                                       | <b>80</b> |
|          | <b>Bibliography</b>   | <b>88</b> |

# List of Publications

The work introduced in this thesis is based on the following publications.

- 
-

# List of Figures

|     |  |    |
|-----|--|----|
| 1.1 | Bag-of-words . . . . .   | 3  |
| 1.2 | Example properties . . . . .   | 3  |
| 4.1 | An example of a hyper-plane in a toy domain of shapes. The hyper-plane for the word square is the dotted line. Green shapes are positive examples and red shapes are negative examples. Those closest to the hyper-plane are less square than those further away . . . . .   | 26 |
| 4.2 | Another example of a hyper-plane in a toy domain of shapes. Here we show multiple directions, one for light and one for square The hyper-plane for the word square is the dotted line. Green shapes are positive examples and red shapes are negative examples for the word square. . . . .  | 31 |
| 4.3 | An example of a Decision Tree classifying if a movie is in the "Sports" genre. Each Decision Tree Node corresponds to a feature, and the threshold T is the required ranking of a document on that feature to traverse right down the tree instead of left. One interesting point to note is that the most important direction is used twice, the "coach, sports, team, sport, football" cluster and results in a majority of negative samples. Another point is that the nodes at depth three are more specific, sometimes overfitting (e.g. in the case of the "Virus" node, likely overfitting to a single movie about a virus) . . . . . | 47 |
| 5.1 | Toy example showing the effect of outliers in a two-dimensional embedding of geometric shapes. . . . .   | 64 |

# List of Tables

|      |  |    |
|------|--|----|
| 3.1  | Text examples from three domains. For the movies and place-type domains, the original text was not available. . . . .                      | 20 |
| 4.1  | Example features from three different domains, where each cluster of words corresponds to a direction which movies are ranked on . . . . . | 28 |
| 4.2  | The top-scoring words for each domain, scoring metric and space type determined by the highest F1-score . . . . .                          | 40 |
| 4.3  | Unique terms between space-types . . . . .   | 42 |
| 4.4  | Different score types . . . . .  | 44 |
| 4.5  | Comparing an MDS sapce to a D2V space for Newsgroups, where a D2V space performed best. . . . .  | 46 |
| 4.6  | summary of all results . . . . .   | 51 |
| 4.7  | Full results for the newsgroups. . . . .   | 54 |
| 4.8  | Results for all other domains for the representations. . . . .   | 55 |
| 4.9  | all dirs . . . . .   | 57 |
| 4.10 | All clustering size results for the newsgroups . . . . .   | 60 |
| 4.11 | The best clustering results for each domain and task . . . . .   | 61 |
| 5.1  | Comparing the highest ranking place-type objects in the original and fine-tuned space. . . . .   | 65 |

---

|     |  |    |
|-----|--|----|
| 5.2 | Results for 20 Newsgroups. . . . .   | 69 |
| 5.3 | The results for Movie Reviews and Place-Types on depth-1, depth-3 and un-<br>bounded trees. . . . .                                  | 70 |
| 5.4 | Results for IMDB Sentiment. . . . .  | 71 |
| 6.1 | Space-types, clusters have the same as single directions. . . . .  | 75 |
| 7.1 | The difference between the representations being directly input to the low-depth<br>decision trees and the word directions . . . . . | 77 |
| 7.2 | Positive Instance Counts for each Class . . . . .  | 79 |

# List of Algorithms

# List of Acronyms

**ML** Machine Learning

**NLP** Natural Language Processing

**NDCG** Normalized Discounted Cumulative Gain

## 0.0.1 Definitions

**Domain** Where the data was originally sourced from  $DOM^I MDB$ , e.g. IMDB movie reviews.

**Word** A string of alphanumeric characters that originated from text in the domain  $DOM_w$ , e.g. the  $w = "Horror"$  from a domain of IMDB movie reviews  $DOM^I MDB$ .

$w$

**Corpus of Documents** A unique group of words, e.g. a review from a domain of IMDB movie reviews  $DOM_I MDB$ .

$C_d w$

**Document** A document of words

$d_w$

**Vector Space** A representation composed of vectors.

$S_v$

**Semantic Space** A representation where spatial relationships between vectors correspond to semantic relationships.

$S_v$



**Word frequency** The frequency of a word  $w$  for its document  $D_w f$ .

$wf$

**Bag-Of-Words** a matrix BOW of documents  $BOW_D$  where each document is composed of unordered frequencies of words  $D = [wf_1, \dots, wf_n]$ . and Conceptual Space we obtain a representation of entities composed of properties. Then, we cover the additional methods we propose to improve this process.

$BOW_d$

**Bag-Of-Words PPMI**

**Feature** A feature is a distinct useful aspect of the domain, corresponding to a numerical value.

$R_f$

**Hyper-plane** The hyper-plane for a word

$H_w$

**Direction vector** The orthogonal direction to a hyper plane that separates a word in a vector space.

$D_w$

**Cluster label** A cluster of words that describe a property.

$C_w$

**Cluster direction** The averaged directions of all words in the label.

$D_C$

**Feature rankings** The rankings induced from a feature direction.

$R_D C$

---

# Chapter 1

## Introduction

Smartphones and other devices that keep people connected to the internet are prevalent, resulting in a deluge of publicly available data. In particular, applications that enable user-generated content e.g. Social Media sites (Facebook, Twitter), Review sites (IMDB, Rotten Tomatoes, Amazon) and content-aggregation sites (Reddit, Tumblr) produce a lot of text data e.g. in posts and comments. Research into how it can be used to solve a variety of problems has been conducted, e.g. using text data to identify if social media posts, or product reviews, are positive or negative [?], to identify social media posts that happen during crises and identify those that are useful to crisis responders [?], or even to predict depression in social media users [?].

The methods to solve these problems are a result of machine-learning, which uses a large amount of data to learn how to solve some specific problem. Machine-learning is a force that has begun to take-part in our day-to-day interactions with both the online and offline space, in the online world through content recommendation, targeted marketing and advertisement, and a variety of businesses that have sprung up with some machine-learning tool at the center. In the offline world, self-driving cars, face recognition on CCTV, behaviour prediction of crowds among others have started taking place across the world. Essentially, these tools are able to succeed because of the availability of data.

Problems that machine-learning can solve can be split into two distinct categories, supervised and unsupervised. Supervised problems have some data that is labelled, and some that is not labelled. The goal of a supervised task is to assign labels to the data that is not labelled, by learning with the data that is labelled. Unsupervised problems do not have any labels, and instead try to solve a problem just from unlabelled data. The learning method to solve the problem is called a machine-learning model. In a supervised task this model learns using the labelled examples, and then is used to label the unlabelled examples.

One example of an unsupervised machine-learning task is to transform data into a representation that can be learned from by the

## 1.1 Representations

Our intuitive understanding of how things behave and what they are in a domain are a representation of reality. This representation is adjusted by our experience with those things in the real world. Each person is constantly adjusting to some degree their representation of what different things are, and this construction and transformation of our personal representations is what we call learning. However, a machine given only text data has no representation. We can understand machines to be numerical reasoners, that can produce things using mathematics from representations. So how can we get a good numerical representation of text so that machines can understand it?

This is the fundamental question we attempt to answer in this section. There are a variety of different ways to achieve a representation of raw text data, and each one is suited for a different task. For example, the task of sentiment analysis can be achieved using a bag-of-words, but as it does not understand context then some things will not be represented well, e.g. a sentence like "This was so good I want to rip my eyes out." although sarcastic, would still count towards the amount of times the word "good" appeared in the document. To achieve better results on this task, it would be better to instead use a representation that can represent context somehow.

## 1.2 Motivation

One task of Natural Language Processing is to obtain this semantic understanding from text by obtaining a machine-readable representation that contains domain knowledge. A basic approach to obtain a representation of this text is to represent entities (e.g. reviews, text-posts) by the frequency of their words, see 1.1.

Below, we show a review with its associated properties labelled.

| <u>Entity: X</u> |                  | <u>Entity: Y</u> |                  | <u>Entity: Z</u> |                  |
|------------------|------------------|------------------|------------------|------------------|------------------|
| <u>Word</u>      | <u>Frequency</u> | <u>Word</u>      | <u>Frequency</u> | <u>Word</u>      | <u>Frequency</u> |
| Dog              | 51               | Dog              | 51               | Dog              | 51               |
| Cat              | 40               | Cat              | 40               | Cat              | 40               |
| Man              | 11               | Man              | 11               | Man              | 11               |
| Cheese           | 0                | Cheese           | 0                | Cheese           | 0                |
| Dog              | 51               | Dog              | 51               | Dog              | 51               |
| Cat              | 40               | Cat              | 40               | Cat              | 40               |
| Man              | 11               | Man              | 11               | Man              | 11               |
| Cheese           | 0                | Cheese           | 0                | Cheese           | 0                |

Figure 1.1: Bag-of-words

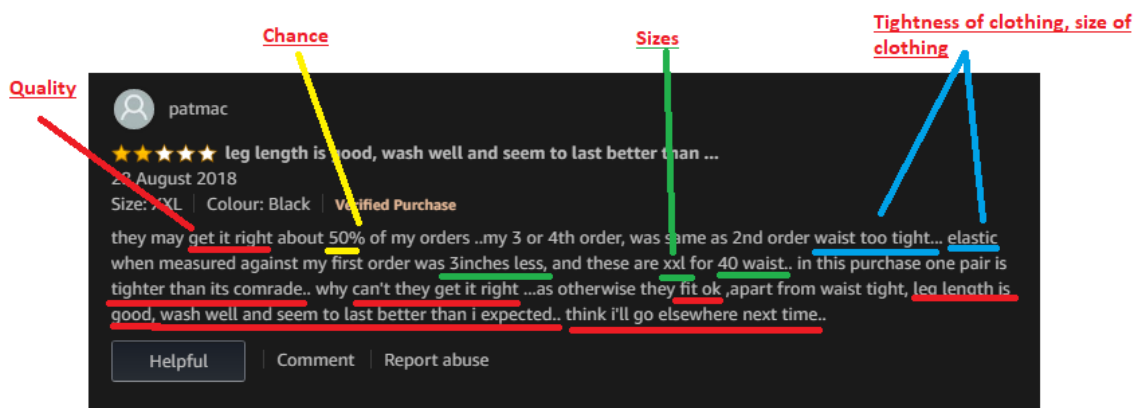


Figure 1.2: Example properties

### 1.2.1 Machine Learning

We can understand these properties to have a degree to which they apply, for example the size of the clothing might be "XXL", "XL", "L", "M" or "S", or the quality may be "Very good", "Good", "Ok", "Bad" or "Very bad". For the former, we may rely on the metadata available

from the site itself, but for the latter the way to obtain this information is less clear. Although we may infer that the rating has some indication of these properties, it does not describe the properties or the degree to which the review refers to them. This kind of information is valuable for making sense of the world of unstructured text, and has broad applications, e.g. The most immediate example is perhaps that they allow for a natural way to implement critique-based recommendation systems, where users can specify how their desired result should relate to a given set of suggestions [33]. For instance, [34] propose a movie recommendation system in which the user can specify that they want to see suggestions for movies that are “similar to this one, but scarier”. If the property of being scary is adequately modelled as a direction in a semantic space of movies, such critiques can be addressed in a straightforward way. Similarly, in [17] a system was developed that can find “shoes like these but shinier”, based on a semantic space representation that was derived from visual features. Semantic search systems can use such directions to interpret queries involving gradual and possibly ill-defined features, such as “*popular* holiday destinations in Europe” [13]. While features such as popularity are typically not encoded in traditional knowledge bases, they can often be represented as semantic space directions.

### 1.2.2 Directions

However, manually labelling these properties and the degrees to which entities (e.g. reviews, text-posts) have them is extremely time-consuming.

A potentially ideal system would be as follows: We collect large amounts of unstructured text data, separated into domains, and obtain the properties of each domain from this data, and rank entities on the degree to which they have these properties. In this way, properties would be understood on a scale built from the domain directly, so that each domain has its own meanings for words according to their own idiosyncrasies. As the process does not require any manual labelling the quality of these properties could be improved simply by obtaining more data. Further, as we are learning from unstructured data, not only would this allow us to understand the data in terms of what we know, but it would also introduce us to new ideas that we may not have previously understood. This kind of representation also has value in application to Machine Learning tasks. If we can separate the semantics of the space linearly into properties,

we are able to learn simple linear classifiers that perform well.

Simple linear classifiers built from a representation composed of rankings on properties have an additional benefit of being more understandable.

## 1.3 Interpretability

Most successful approaches in recent times, like vector-spaces, word-vectors, and others, rely on the distributional model of semantics. This model relies on encoding unstructured text e.g. of a movie review, as a vector, where each dimension corresponds to how frequent each word is, we are able to calculate how similar the entities are, e.g. we know that if two movies have a similar distribution of words in their reviews, like frequent use of the word 'scary', or 'horror', then they would have a higher similarity value. These models, also known as 'semantic spaces' encode this similarity information spatially.

Semantic relationships can be obtained from semantic spaces.

applications/need for good interpretability:

- Safety
- Troubleshooting, bug fixing, model improvement
- Knowledge learning
- EU's "Right to explanation"
- Discrimination

properties of an interpretable classifier:

- Complexity: 'the magic number is seven plus or minus two' [28] also has many positive effects for its users, like lower response times [23, 11], better question answering and confidence for logical problem questions [11] and higher satisfaction [23].
- Transparency:

- Explainability:
- Generalizability:

Properties, entities, the benefits and application of a representation formed of these

Basic introduction to directions, explanation of the utility and application of our approach

## 1.4 Thesis Overview / Contributions

In ??, we focus on further experimenting with one relationship that was formalized in [6]: a ranking of entities on properties. In particular, we use this method of building a representation of entities as a way to convert a vector space into an interpretable representation, for use in an interpretable classifier. The reason that we chose this representation to expand on is because by representing each entity  $e$  with a vector  $v$  that corresponds to a ranking  $r$ , the meaning of each dimension is distinct, and we are able to find labels composed of clusters of words for these dimensions. Here, we make the distinction between a property and a word, a property is a natural property of the space that exists in terms of a ranking of entities, and words are the labels we use to describe this property.

---

## Chapter 2

# Background

## 2.1 Introduction

This section focuses on providing background and context for the specific problems that this thesis solves. It begins with a gentle introduction to some terminology and the general flow of how problems are solved using machine-learning and text representations.

Not all rules that govern the meaning of text data are universal. The text data available from a domain is unique to that domain in many ways, for example on the social media site Facebook text data can be formatted into posts and comments, and is posted by users. Although a movie review by a critic could on the surface have the same structure - a post with comments below, the rules governing what is contained in the post are clearly not the same. One goal of machine-learning is to predict if a piece of text will be shared or liked by users. In this case, it is clear that in-order to determine if a movie review will be liked or shared, it is difficult to determine if that will be the case when using the same logic that you would for a facebook post. Facebook posts that trend or are well-liked are typically brief, easily consumed and focused on humour. Meanwhile, movie reviews that trend are due to cutting and intelligent analysis of a movie in a relatable way.

The main point here is not that these domains are completely different, but rather to achieve good results on some particular task in the domain, it is important to make the distinction between text from the domain and text outside of it. Of-course, although there are clearly different rules for each domain, that does not mean that there are not common trends between them. Typically in machine-learning the methods that perform the best use some-kind of large-scale data that is not directly related to the domain as well as a lesser amount of domain data. This thesis focuses on solving domain-specific problems.



The corpus of text obtained from a domain is usually arranged to reflect the task to be solved. For example, in a machine-learning task that attempts to determine what genre a movie is automatically using the text from movie reviews, the corpus would be formatted such that all the reviews for one movie would be used in the same document. Arranging the corpus into documents like this from raw text is necessary when attempting to determine the differences between objects in the domain, like movies in a domain of movie reviews, or people in the domain of facebook. Generally, a distinction is made between document-based tasks and other tasks. In the case of this thesis, we look at document-based domain-specific tasks.

There are a variety of ways to add additional structure to raw text, one such way is to label parts-of-speech (known as POS tagging) like nouns, adjectives, or other grammatical constructs. This can be done automatically with reasonable accuracy, however there are not many datasets with this kind-of structured information available, and it is difficult to achieve reliable results without experts annotating the data, which is costly and time-consuming. This thesis focuses on how to use raw text without adding additional structure or information. Using raw text without additional structure enables the method will have broad applicability and allows easy comparison with other work.

Humans can have an intuitive understanding of the semantics that are present in unstructured text, but machines do not. To obtain a machine intelligible representation of raw text it is not necessary to annotate grammar and meaning symbolically, rather a simple representation that can scale to an extreme amount of data performs well. When looking to achieve state-of-the-art results, representations that are more complex are usually used. However although a representation may be simple, with enough examples even a basic representation usually has enough information to clearly distinguish between types of objects in a domain for a task.

One of the most common representations is a bag-of-words (BOW) which ignores word context, instead taking the words that occur in each document and assigning a value to them in a matrix, e.g. where each word in a document is assigned a value for its frequency in that document. For example, a short document of text like "there was a dog, and a man, and the man, and the dog" would be translated into word frequencies "there: 1, was: 1, a: 2, and: 3, the: 2, man: 2, dog: 2". This representation is simple, and ignores word context, grammar and punctuation but is highly effective when using machines to solve problems using a large amount of unstructured text documents. This thesis uses a representations that rely on the bag-of-words in some form,

despite its simplicity.

Representations are used to learn how to separate different kinds of objects in a domain. This is called a classification problem which can be described as follows: given some labelled documents  $D_l$  (e.g., movie reviews labelled as having positive sentiment or negative sentiment), predict the label of an unlabelled document  $d$ . If the classifier performs well, and can predict a variety of unlabelled documents  $d = d_l$ , the representation must somehow contain the knowledge of what sentiment is. In the case of a bag-of-words, we can understand that the frequencies of sentiment-related words, like "good", "bad", and "thrilling" would be good enough to achieve reasonable performance, as a machine-learning classifier could determine rules based on the frequency of these relevant words, e.g. "IF good > 30, and thrilling > 20, THEN positive sentiment". The tasks that are solved in this thesis are all classification tasks.

Simple rules are a good basis for an interpretable classifier. However, this is only possible when the features (e.g. the word frequencies in a bag-of-words) are clearly defined. This forms the basis of what an interpretable representation is viewed as in this thesis. If the features correspond to some meaningful property in the domain, e.g. "good" or "thrilling", then that is treated as an interpretable representation. In this sense, a bag-of-words is an interpretable representation, however the problem solved in this thesis is about achieving a representation that uses complex relations but is still interpretable.

This Chapter informs the reader of the fundamentals behind obtaining a representation from text data and using it to solve machine-learning problems, as well as explaining how interpretable representations are used in the literature. Section 2.2.1 covers data preprocessing and management in preparation for obtaining a representation. Then, Section ?? describes the process of obtaining different kinds of representations from this data. To complete a basic pipeline, Section 2.3 covers different machine-learning methods to solve problems. Finally, interpretable representations and classifiers are covered in Section ?? to give context in the literature for the work in the next three Chapters.

## 2.2 Text Representations

In the next Chapter of the thesis, as well as in section ?? we discuss how to make a representation that both humans and machines can understand, but this section focuses on representations that are useful to machines when used for machine-learning, rather than being interpretable.

### 2.2.1 Preprocessing

To obtain a representation of a corpus, the text data must first be processed so that the representation does not contain any superfluous information - or "noise" that would disrupt it. What exactly noisy data is depends on the representation and the task, but for this thesis it can be seen as parts of the text data that are not meaningful when distinguishing between types of objects in the domain. For example, a word starting with an uppercase or lowercase letter is probably not relevant for determining the genre of a movie from its movie review text. This is noise that if retained in the representation would harm it more than help it. Another example of noise that you would ideally like to remove is metadata, e.g. if the email of a movie reviewer was retained in their review text, that will not be useful information for a task related to the movie.

Specifically, when representing for a domain a way to begin is to build a vocabulary  $V_w$ , composed of unique words  $w \in V$ . In this vocabulary, it is important that words which have the same meaning are not treated as different words e.g. if the word "Dog" was considered to be different to the word "dog." then the vocabulary would be too noisy. There are standard methods for removing noise in a dataset. We describe them in the following bullet-points:

- Convert all text to lower-case, e.g. "The man and The Dog" converted to "the man and the dog"
- Remove all punctuation including excess white-space, e.g. "the man, and the dog..." converted to "the man and the dog"
- Using a predefined list of "stop-words", listed in full in Table ??, remove words that are not useful, e.g. "the man and the dog" converted to "man dog"
- Remove infrequent words, e.g. "man dog, dgo, dog man" converted to "man dog, dog man".

- Domain-specific pre-processing to remove metadata, e.g. removing emails from the end of movie reviews.

It can be assumed that for all of the methods that are described in this section, the rules above are applied to the corpus beforehand. In terms of removing words that were not frequent enough, words that did not occur more than once are removed. Although these rules are not universal, they are a good basis for computational methods of representing text data that do not rely on word-context and grammar. In the next section, we cover a variety of methods for text representation and explain their basic utilities.

### 2.2.2 Bag-of-words

As mentioned in the previous section, unnecessary parts of the data that are not meaningful for the task should be removed. The bag-of-words is a representation that comes with the following assumption: the context of words does not need to be retained to perform well on the task. How correct this assumption is depends on the task, but despite this view being overly-simplistic the application and use of the bag-of-words (BOW) is broad. There are multiple ways to represent words in the BOW format, but the most common is by the frequency of the words in a document.

The natural structure for this kind of representation is that of a matrix, where rows are documents and columns are words in the domain as defined by their vocabulary. Specifically, text documents in a domain  $d \in D$  have an associated vocabulary of unique words across all documents  $w \in W$ . The bag-of-words  $B_D$  is a matrix where each document is a row, and each column is a word, where the value of each word for a document is the word's frequency in that document  $d = (wf_1, wf_2, \dots, wf_n)$  where  $wf(d)$  is equal to the frequency of a word in a document and  $n$  is equal to the number of unique words in the vocabulary for all documents  $w \in W$ .

#### Term Frequency Inverse Document Frequency (TF-IDF)

There are two main problems of using frequency is that words which are frequent in the domain are given a higher value than words which are used frequently only in a single document. First, longer documents result in overall higher values than shorter ones. Second, words that are

frequent in many documents are given equal importance to those that are frequent only in some documents. However, we are concerned with what distinguishes documents from each other so giving equal importance for example, in the domain of movie-reviews, to the word "movie" does not accurately represent how important it is for the meaning of the movie. Rather, we would be interested in terms that are frequent for only that movie review, as for example if the term "gore" was frequent in only five different movies out of 15,000 then it is clearly important for those movies.

The idea that words which are infrequent overall but frequent for some documents are important can be applied to a bag-of-words using the Term Frequency Inverse Document Frequency (TF-IDF) formulae. The first part of TF-IDF is Term Frequency  $TF_{d,w}$ , which is a normalization of frequency that solves the first problem of larger documents being treated as more important than shorter ones.

$$TF_{(w,d)} = \frac{wf(d)}{\sum_n wf_n(d)}$$

Where  $wf(d)$  is the number of occurrences of word  $w$  in document  $d$  and  $n$  is the number of words overall in the vocabulary. The next part of TF-IDF is Inverse Document Frequency, which is a measure that rewards terms that have a low Document Frequency.

$$IDF_w = \frac{d_n}{df(w)}$$

Where  $df(w)$  is the amount of documents the word  $w$  has occurred in and  $d_n$  is the amount of documents in the corpus. Note that while Term Frequency measures the frequency of a term in a document relative to that documents length, Document Frequency measures the overall occurrences of the term across all documents, relative to the number of documents. Essentially, it measures how rare that term is for a document, rather than how rare it is for a word. Finally, the TF-IDF is just the Term Frequency multiplied by the Inverse Document Frequency.

$$TF - IDF = TF \times IDF$$

### Positive Pointwise Mutual Information (PPMI)

Positive Pointwise Mutual Information (PPMI): Follows the same idea as TF-IDF. PPMI is defined as

$$ppmi(w, d) = \max \left( 0, \log \left( \frac{p_{wd}}{p_{w*} \cdot p_{*d}} \right) \right)$$

where

$$P_{wd} = \frac{wf(d)}{\sum_{w'} \sum_{d'} wf'(d')}$$

where  $wf(d)$  is the number of occurrences of word  $w$  in document  $d$ , and

$$P_{w*} = \sum_{d'} P_{wd'} \qquad P_{*d} = \sum_{w'} P_{w'd}$$

### 2.2.3 Principal Component Analysis

Principal Component Analysis is a dimensionality reduction method that results in dimensions ordered by importance. In application to text-processing, this matrix could be the document by word matrix of PPMI values. Essentially, we want to go from the extremely large and sparse PPMI matrix to a dense matrix of our own specified size. To do so, we can apply singular value decomposition (SVD) of a matrix that has been normalized. This method can only model linear relationships.

### 2.2.4 Multi-Dimensional Scaling

The goal of multi-dimensional scaling is to create a space that spatially represents the dissimilarity between documents. So if two text documents are very dissimilar, they will be spatially distant from each other. The same as PCA, this is a dimensionality reduction where the amount of dimensions are specified, but it requires a  $D_n \times D_n$  dissimilarity matrix, which can consume a lot of memory when being applied. Instead of PCA, where dimensions are ordered by importance, the resulting dimensions of MDS are not as clearly meaningful. However, this method can model non-linear relationships.

## 2.3 Classification and Regression

When using a bag-of-words as a representation in a classification task, that threshold is usually set higher, as it is more important to remove noise that would not naturally be removed when using the bag-of-words to create another representation.

Classification of text documents can be used for example to identify if social media posts, product reviews, etc are positive or negative [?], identify social media posts that happen during crises and automatically categorize them to be useful to responders [?], or detect infections acquired while patients are in a hospital . However, text documents like news articles, product reviews or social media posts cannot be classified without first being represented computationally. Representations  $r$  are composed of features  $r = (x_1, x_2, \dots, x_n)$ , where ideally each feature  $x$  is meaningful in the domain. For example, meaningful features when determining the value of a house would be the amount of bedrooms  $x_1$ , and the amount of toilets  $x_2$ . An example vector from these examples would be  $[5, 2]$  for a house with 6 bedrooms and 3 toilets.

One way to obtain features for text documents is to use the frequencies of words in that document. As a vector  $d = (wf_1, wf_2, \dots, wf_n)$  with  $wf_1$  referring to first word in the vocabulary  $w \in W$ , and  $wf_n$  referring to the final word in the vocabulary. This is called a Bag-Of-Words (BOW), called as such because word-order is not retained. However, to have a consistent bag-of-words representation, the text must be normalized so that any word  $w \approx w$  will  $w = w$ , so where a word varies in format but not alphanumeric characters it is treated as the same word, e.g. "Wow, wow, WOW!!!" would be treated as "wow wow wow". This is a common step taken when producing a representation from text, where it is simplified to make it easier to represent.

In-order to classify documents, a label  $y$  is required. Labels can be understood as categories in the domain, e.g. in the domain of sentiment analysis on movie reviews, labels could be "very good", "good", "average", "bad", "very bad" represented as  $[0, 1, 2, 3, 4]$ . Given features  $x$  and labels  $y$ , a model  $m$  learns a way to predict the label of a document given its features, and this learned method can be applied to unlabelled documents. For example, given a bag-of-words representation, one way to automatically label a news article category would be with thresholds on the frequency of words in the text e.g. if the word "amazing" and the word "great" both occur more frequently than a threshold  $T$  determined by a model, then the label is 0, a "very good" movie.

### 2.3.1 Evaluation Metrics

To evaluate a model, the difference between the real labels of documents and the predicted features of documents is compared. However, the value of the model is in its ability to predict the labels of documents that are unlabelled. Typically, this problem is solved by splitting the documents into a training set and a test set. The training set is used when learning the model, and the test set is used to verify the model is working correctly.

Here, we assume we are classifying a single binary class, where positive labels are 1 and negative labels are 0. The most simple way to evaluate a model is by its accuracy  $a$ , where  $t_n$  is the number of correct predictions, and  $P_n$  is the number of all predictions.

$$a = t_n / P_n$$

However, this can give a misleadingly high score if for example, the dataset is unbalanced with many more negative labels than positive ones, and the model predicts only negatives. An example of where this would be the case is when classifying out of all social media posts, which ones are important for emergency responders to investigate. Although there are very few positive instances of this class, identifying those is very important. In the case of a model predicting only negatives, the accuracy would be high as the number of correctly predicted negatives  $tn$  is high, but the model has not actually learned anything, which we can tell by looking at the number of correctly predicted positives  $tp$ . For a metric that can take this into account, we must consider the number of incorrectly predicted positives (negatives classified as positive)  $fp$  and the number of incorrectly predicted negatives  $fn$ .

In this situation, the metric we would want to optimize would be recall. Recall  $rec$  is the proportion of true positives  $tp$  identified correctly.

$$rec = tp / (tp + fn)$$

In the case of a model predicting only negatives, the  $rec$  would be zero. Recall is useful in these situations where we are interested in how many false negatives  $fn$  there are. However, if the model is instead prioritizing positive predictions too much rather than negative ones, we can use precision  $pre$



$$pre = tp/tp + fp$$

### 2.3.2 Decision Trees

Decision Trees are a model that produce a tree of decisions, composed of nodes. Each node has its own feature and associated threshold value  $T$ . If the value given in the feature is larger than the threshold  $T$ , then it traverses one direction, otherwise it traverses the other direction. Eventually, upon reaching the leaf node the decision made on the threshold is the classification of the document. Decision Trees work well as they are simple, and easy to understand. However, to model most complex relationships in e.g. an MDS space, there would need to be many nodes to achieve a strong classification result, as the features are not necessarily meaningful independently. Additionally with a vector space, it is not easy to understand, as the features used are not clearly meaningful. Ideally we would want clearly labelled features like in a bag-of-words, but to model complex domain tasks this would require many nodes, which greatly increases complexity of the tree.

## 2.4 Neural Networks

### 2.4.1 Feedforward Neural Networks

### 2.4.2 Word Vectors

### 2.4.3 Doc2Vec

## 2.5 Support Vector Machines

- Performance increase for support vector machines on sparse data, balancing, etc
- C parameters, gamma parameters

## 2.6 Clustering

### 2.6.1 K-means

### 2.6.2 Derrac's K-means Variation

## 2.7 Interpretable Representations

### 2.7.1 Topic Models

## 2.8 Interpretable Representations

a. NNSE b. compositional c. 2007 paper as wikipedia similarities d. Topic models e. Infogan, etc

[36] Sparse PCA (Why not compare lol)

Vector space models typically use a form of matrix factorization to obtain low-dimensional document representations. By far the most common approach is to use Singular Value Decomposition [4], although other approaches have been advocated as well. Instead of matrix factorization, another possible strategy is to use a neural network or least squares optimization approach. This is commonly used for generating word embeddings [22, 25], but can similarly be used to learn representations of (entities that are described using) text documents [2, 31, 13]. Compared to topic models, such approaches have the advantage that various forms of domain-specific structured knowledge can easily be taken into account. Some authors have also proposed hybrid models, which combine topic models and vector space models. For example, the Gaussian LDA model represents topics as multivariate Gaussian distributions over a word embedding [3]. Beyond document representation, topic models have also been used to improve word embedding models, by learning a different vector for each topic-word combination [19].

The most commonly used representations for text classification are bag-of-words representations, topic models, and vector space models. Bag-of-words representations are interpretable in principle, but because the considered vocabularies typically contain tens (or hundreds) of

thousands of words, the resulting learned models are nonetheless difficult to inspect and understand. Topic models and vector space models are two alternative approaches for generating low-dimensional document representations.

### **2.8.1 Word Vectors**

---

## Chapter 3

# Datasets and Semantic Spaces

## 3.1 Introduction

For the experiments in this thesis, five different domains are used, each with their own particular vocabulary and meaning of words in their vocabulary. This Chapter begins with a section to give insight into the datasets with explanations of each domain, accompanying examples, and their classes. This is followed by technical descriptions of preprocessing methods for the datasets. Finally, we introduce the bag-of-words and semantic space representations built from these preprocessed datasets that will be used in the remainder of the thesis.

## 3.2 Datasets

First, we go through the history and class names of the datasets to give context, and provide examples of unprocessed text from three domains in Table 3.1.

**IMDB Sentiment** Where documents are exclusively highly polar IMDB movie reviews, either rated  $\leq 4$  out of 10 or  $\geq 7$  out of 10. Reviews were collected such that it was limited to include at most 30 reviews from any movie in the collection, as some movies contained many more reviews than others. The corpus is split half and half between positive and negative reviews, with the task being to identify the sentiment of the review.

**20 Newsgroups**<sup>1</sup> Originating from online news discussion groups from 1995 called newsgroups, where group email-type discussions are made by users about particular topics within 20 different groups. In this dataset, each document is composed of a topic, where user posts are concatenated

---

<sup>1</sup><http://qwone.com/~jason/20Newsgroups/>

| Data Type  | Unprocessed  | Processed  |
|------------|--|--|
| Newsgroups | morgan and guzman will have era's 1 run higher than last year, and the cubs will be idiots and not pitch harkey as much as hibbard. castillo won't be good (i think he's a stud pitcher)   | morgan guzman eras run higher last year cubs idiots pitch harkey much hibbard castillo wont good think hes stud pitcher  |
| Sentiment  | All the world's a stage and its people actors in it--or something like that. Who the hell said that theatre stopped at the orchestra pit--or even at the theatre door? Why is not the audience participants in the theatrical experience, including the story itself?<br>This film was a grand experiment that said: "Hey! the story is you and it needs more than your attention, it needs your active participation". "Sometimes we bring the story to you, sometimes you have to go to the story."<br>Alas no one listened, but that does not mean it should not have been said." | worlds stage people actors something like hell said theatre stopped orchestra pit even theatre door audience participants theatrical experience including story film grand experiment said hey story needs attention needs active participation sometimes bring story sometimes go story alas one listened mean said |
| Reuters    | U.K. MONEY MARKET SHORTAGE FORECAST REVISED DOWN The Bank of England said it had revised its forecast of the shortage in the money market down to 450 mln stg before taking account of its morning operations. At noon the bank had estimated the shortfall at 500 mln stg.  | uk money market shortage forecast revised bank england said revised forecast shortage money market 450 mln stg taking account morning operations noon bank estimated shortfall 500 mln stg   |

**Table 3.1: Text examples from three domains. For the movies and place-type domains, the original text was not available..**

together. The groups that topics are categorized by are Atheism, Computer Graphics, Microsoft Windows, IBM PC Hardware, Mac Hardware, X-Window (GUI Software), Automobiles, Motorcycles, Baseball, Hockey, Cryptography, Electronics, Medicine, Space, Christianity, Guns, The Middle East, General Politics and General Religion, which also act as the classes for this dataset when being evaluated. Generally, it can be quite easy to identify if a document belongs to a particular group if it uses a keyword unique to that group, e.g. the word "chastity" will almost always mean that the document belongs to the "Christianity" class.

**Reuters-21578, Distribution 1.0** Text from the Reuters financial news service in 1987, composed of a headline and body text. The classes were chosen with assistance from personnel at reuters<sup>2</sup>, meaning that they can contain jargon. For that reason, explanations are provided with the original names in brackets. The classes are Trade, Grain, Natural Gas (nat-gas), Crude Oil (crude), Sugar, Corn, Vegetable Oil (veg-oil), Ship, Coffee, Wheat, Gold, Acquisitions (acq), Interest, Money/Foreign Exchange (money-fx), Soybean, Oilseed, Earnings and Earnings Forecasts (earn), BOP, Gross National Product (gnp), Dollar (dlr) and Money-Supply.

<sup>2</sup>For more detail on the history of the dataset: <https://archive.ics.uci.edu/ml/datasets/reuters-21578+text+categorization+collection>

**Placetypes** Taken from work by Derrac [6]. Originating from the photo-sharing website flickr, where photos are tagged (i.e. words describing the photos like "sepia" or "mountain") by users. 22,816,139 photos were considered, and tags that occurred in place-type taxonomies (GeoNames, a taxonomy of man-made and natural features, Foursquare a mostly flat taxonomy of urban man-made places like bars and shops, and the site category for the common-sense knowledge base taxonomy OpenCYC) with more than 1,000 occurrences were chosen as documents. Each document, named after a flickr tag, is composed of all flickr tags where that tag occurred. There are three tasks, generated from the three different place type taxonomies. The Foursquare taxonomy, classifying the 9 top-level categories from Foursquare in September 2013, Arts and Entertainment, College and University, Food, Professional and Other Places, Nightlife Spot, Parks And Outdoors, Shops and Service, Travel and Transport and Residence. the GeoNames taxonomy limited to 7 classes, Stream/Lake, Parks/Area, Road/Railroad, Spot/Building/Farm, Mountain/Hill/Rock, Undersea, and Forest/Heath, and the OpenCYC Taxonomy, which we limited to 25 classes, Aqueduct, Border, Building, Dam, Facility, Foreground, Historical Site, Holy Site, Landmark, Medical Facility, Medical School, Military Place, Monsoon Forest, National Monument, Outdoor Location, Rock Formation, and Room. Naturally as these tasks were derived from taxonomies they are multi-label.

**Movies** Taken from work by Derrac [6]. The top 50,000 most voted-on movies were chosen for this dataset initially, and reviews were collected from four different sources (Rotten Tomatoes, IMDB, SNAP project's Amazon Reviews <sup>3</sup> and the IMDB sentiment dataset. Then, the top 15,000 movies with the highest number of words were chosen as documents, where each document is composed of all of that movies reviews concatenated together. Three tasks are used to evaluate this dataset: 23 movie genres, specifically Action, Adventure, Animation, Biography, Comedy, Crime, Documentary, Drama, Family, Fantasy, Film-Noir, History, Horror, Music, Musical, Mystery, Romance, Sci-Fi, Short, Sport, Thriller, War, Western. 100 of the most common IMDB plot keywords (See Appendix ??) and Age Ratings from the UK and US, USA-G, UK-12-12A, UK-15, UK-18, UK-PG, USA-PG-PG13, USA-R.

<sup>3</sup><https://snap.stanford.edu/data/web-Amazon.html>.

### 3.3 Technical Details

In this section, we describe the vocabulary and document sizes for each domain. Each domain is preprocessed such that it is converted to lower-case, non-alphanumeric characters are removed and whitespace is stripped such that words are separated by a single space. Words are removed from a standard list of English stop-words from the NLTK library [?] and we filter out terms that do not occur in at least two documents, with an additional limit to the maximum number of words in a vocabulary set to 100,000.

**IMDB Sentiment**<sup>4</sup> When the original corpus was produced, the 50 most frequent terms were removed. It contains 50,000 documents with a vocabulary size of 78,588. After removing terms that did not occur in at least two documents, the vocabulary size was reduced to 55384. the number of positive instances in the classes is 25,000.

**20 Newsgroups**<sup>5</sup> Obtained from scikit-learn. <sup>6</sup> Originally containing 18,846 documents, in this work it is preprocessed using sklearn to remove headers, footers and quotes. Then, empty and duplicate documents are removed, resulting in 18302 documents. The vocabulary size (unique words) is 141,321. The data is not shuffled. After filtering out terms that did not occur in at least two documents, we end up with a vocabulary of size 51,064. This is a larger change than the sentiment dataset, despite beginnging with a larger vocabulary, likely because newsgroups contains many terms that were not relevant to a majority of the documents, instead being particular to their groups. The number of positive instances averaged across all classes is 942, around 5%.

**Reuters-21578, Distribution 1.0** Obtained from NLTK<sup>7</sup> originally containing 10788 documents. After removing empty and duplicate documents the result is 10655 documents. Originally contained 90 classes, but as they were extremely unbalanced all classes that did not have at least 100 positive instances were removed, resulting in 21 classes. The original vocabulary size is 51,001 and all words that did not occur in at least two documents were removed, resulting in a vocabulary size of 22,542. The number of positive instances averaged across all classes is 541, around 5%.

---

<sup>4</sup>Obtained by: <https://keras.io/datasets/>, Originally from <https://ai.stanford.edu/amaas/data/sentiment/> [20]

<sup>5</sup><http://qwone.com/~jason/20Newsgroups/>

<sup>6</sup>[https://scikit-learn.org/0.19/modules/generated/sklearn.datasets.fetch\\_20newsgroups.html#sklearn.datasets.fetch\\_20newsgroup](https://scikit-learn.org/0.19/modules/generated/sklearn.datasets.fetch_20newsgroups.html#sklearn.datasets.fetch_20newsgroup)

<sup>7</sup><https://www.nltk.org/book/ch02.html>

**Placetypes** It originally has a vocabulary size of 746,527 and 1383 documents. This is a very large vocabulary size to document ratio. The end vocabulary for this space was of size 100,000 due to the hard limit. This is roughly equivalent to removing all documents that would not be in at least 6 documents. As most classes in this domain are extremely sparse (less than 100 positive instances) no classes are deleted. As 8 of these remaining classes had a low number of positive occurrences, OpenCYC classes are removed that do not have positive instances for at least 30 documents, leaving us with 17. For the Geonames taxonomy, the same rule resulted in only 7 of 9 categories being used.

**Movies** Another large dataset with a vocabulary size of 551,080 and a document size of 15,000. However, after investigating the data made available by the authors, it was found that there were a number of duplicate documents. After removing these duplicate documents, there are 13978 documents. In the same way as the place-types, the vocabulary hit the hard limit of size 100,000.

## 3.4 Representations

For the bag-of-words representation used as a baseline, terms are additionally filtered out that do not occur in at least 0.001% of documents, as to scale with the amount of documents in each domain. From this filtered vocabulary, a bag-of-words is obtained by creating a matrix of documents and words, with the values of that matrix corresponding to how frequent each word was for each document. However, as frequency bag-of-words are not able to distinguish between frequent terms (e.g. "the") and important terms, words are weighed such that words which occur frequently in a small amount of documents are given a higher value than those that occur frequently in a large amount of documents. To do this, Positive Pointwise Mutual Information (PPMI) scores are used, following success in similar work by [6]. See section ?? for more detail.

For the work in the following chapters, we wanted a variety of different Vector Space Models. Below the choices for the Vector Space Models that are formally described in Section ?? are explained:

**Multi-Dimensional Scaling (MDS):** Following [5], we use Multi-Dimensional Scaling (MDS) to learn semantic spaces from the angular differences between the PPMI weighted BoW vectors.



This was chosen as it performed well in previous work by [6], and is a non-linear transformation based on PPMI vectors.

**Principal Component Analysis (PCA):** We use PCA as a linear transformation of the PPMI weighted BoW vectors, as it is a standard dimensionality reduction technique used historically and prevalently today to serve as a baseline reference.

**Doc2Vec (D2V):** Inspired by the Skipgram model [18]. A distributional document representation used as a representative of a higher performing method of learning in terms of document classification. For the Doc2Vec space, the hyper-parameters are additionally tuned for the *window size*(5, 10, 15) referring to the context window, the *mincount*(1, 5, 10) referring to the minimum frequency of words and the *epochs*(50, 100, 200) of the network for each size space. We chose the best parameters for each class in each domain by evaluating the space as input to linear SVM's, which were tuned with two parameters: the best C values 1.0, 0.01, 0.001, 0.000] and if the weights should be balanced such that positive instances are weighted in proportion to how rare they are.

**Average Word Vectors (AWV):** Finally, we also use semantic spaces that are composed of averaged word vectors, using a pre-trained GloVe word embeddings trained on the Wikipedia 2014 + Gigaword 5 corpus<sup>8</sup>. While simply averaging word vectors may seem naive, this was found to be a competitive approach for unsupervised representations in several applications [10]. For each document, we simply average the vector representations of the words that appear at least twice in the BoW representation.

---

<sup>8</sup><https://nlp.stanford.edu/projects/glove/>

# Re-organizing Vector Spaces into Interpretable Representations

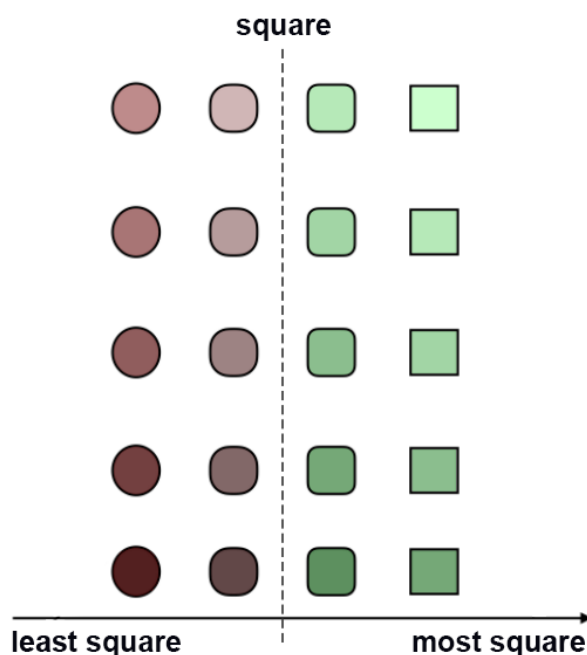
## 4.1 Introduction

Feature-directions = directions that make good features Our argument is: directions that make good features are cluster-directions

Vector space models encode meaning spatially, but their features are typically uninterpretable. This lack of interpretability limits their off-the-shelf application in real-world domains like Medicine, the Criminal Justice System and Financial Markets (discussed in Section 2.8). However, they achieve strong results in a variety of domains and see widespread use as they are flexible in how they can be learned, e.g. by integrating word-context to achieve strong results on sentiment tasks [26], learning visual data alongside word-data to explain the content of images [21], and enforcing grammatical structure to perform better at question answering tasks [24].

This chapter is about re-structuring vector-space representations such that their features correspond to interpretable spatial structures in the original vector space. To give insight into what kind-of features this method can obtain, we can give an example from a domain where documents are concatenated movie reviews for a particular movie (See Section 3.2). In this domain, documents would be represented by features like "Scary", which would be how scary a movie is, or "Romantic" which would be how romantic a movie is.

This chapter follows work by Derrac[6], who first introduced the method to achieve this. The method begins with the following assumption: if a vector space can be linearly separated such



**Figure 4.1:** An example of a hyper-plane in a toy domain of shapes. The hyper-plane for the word square is the dotted line. Green shapes are positive examples and red shapes are negative examples. Those closest to the hyper-plane are less square than those further away.

that documents where a word occurs are separated from those where that word does not occur, that word is semantically important in the domain. This can be achieved in an unsupervised way by training a linear model, e.g. a linear Support Vector Machine (SVM) (See Section 2.5), to separate documents for words in a bag-of-words, where if a word occurs in a document it is treated as a positive example, and if it does not then it is a negative example. Then, the words that are most semantically important in the domain can be determined by evaluating how well the documents are separated using standard model evaluation metrics like F1-score (see Section ??).

In the case of a linear SVM, for each model trained on a word a hyper-plane is obtained separating documents that contain the word and documents that do not contain it. We show an example of this in a toy domain of shapes in 4.1, where the dotted line is a hyper-plane. As shown in this example, it can be assumed that documents furthest from the hyperplane on the negative side are the least representative of the thing being separated by the hyper-plane, in this

case the 'squareness' of a shape, and the documents that are furthest from the hyper-plane on the positive side are the most representative, while those closest to the hyper-plane are more ambiguous. Given this hyper-plane, a direction can be obtained that goes from documents that are the most distant from the hyper-plane on the negative side, to those that are most distant from the hyper-plane on the positive side by simply taking the orthogonal vector (the direction shown at the bottom of the example in 4.1).

By measuring how far up a document is on the orthogonal direction vector for a word, we can obtain a ranking of that document on that word, e.g. how 'Scary' it is relative to the other documents. After repeating this process for all documents on a word direction, we can obtain a ranking of all documents on a word that can be used as a feature for the representation. This forms the basis of our approach towards restructuring a vector space such that the features encoded spatially are used directly as features of the representation. We first obtain hyper-planes for all words based on binary frequency. Then, we identify words which are semantically important (by e.g. F1 Score or accuracy of the hyper-plane). Finally, we obtain orthogonal directions for these word hyperplanes and rank documents on how far up they are on these directions. This ranking is then used as a feature of the new representation.

However, it can be sometimes unclear what a document being high up on a direction can mean when it's just a single word. For example, in a domain of IMDB movie reviews "numbers" could be referring to musical "numbers" in a Broadway musical movie, or the amount of mathematics done by the actors. To resolve this, similar words can be clustered together e.g. we can give context to the word "numbers" by clustering together similar word directions "singing songs musical song numbers dance dancing sings sing Broadway". This can be done with an off-the-shelf clustering algorithm like K-means (see 2.6). We show examples of these word cluster labels for the features of our interpretable representation in 4.1.

Directions in vector spaces that go from documents that least represent a word, to those that most represent it, can be useful in a wide variety of applications. The most immediate example is perhaps that they allow for a natural way to implement critique-based recommendation systems, where users can specify how their desired result should relate to a given set of suggestions [32]. For instance, [35] propose a movie recommendation system in which the user can specify that they want to see suggestions for movies that are "similar to this one, but scarier". If the direction of being scary is adequately modelled in a vector space of movies, such critiques can

| IMDB Movie Reviews                        | Flickr-Placetypes                    | 20-Newsgroups                        |
|---|--------------------------------------|--------------------------------------|
| courtroom legal trial court               | broadway news money hollywood        | switzerland austria sweden swiss     |
| disturbing disgusting gross               | fir bark activism avian              | ham amp reactor watts                |
| tear cried tissues tears                  | palace statues ornate decoration     | karabag armenian karabakh azerbaijan |
| war soldiers vietnam combat               | drummer produce musicians performers | 4800 parity 9600 bps                 |
| message social society issues             | ubahn railways electrical bahn       | xfree86 linux                        |
| events accuracy accurate facts            | winery pots manor winecountry        | umpires umpire 3b viola              |
| santa christmas season holiday            | steeple religion monastery cathedral | atm hq ink paradox                   |
| martial arts kung                         | blanket whiskers fur adorable        | lpt1 irq chipset mfm                 |
| bizarre weird awkward                     | desolate eerie mental loneliness     | manhattan beauchaine bronx queens    |
| drug drugs dealers dealer                 | carro shelby 1965 automobiles        | photoshop adobe                      |
| inspirational inspiring fiction narrative | relax dunes tranquil relaxing        | reboost fusion astronomers galactic  |

**Table 4.1: Example features from three different domains, where each cluster of words corresponds to a direction which movies are ranked on.**

be addressed in a straightforward way. Similarly, in [16] a system was developed that can find “shoes like these but shinier”, based on a vector space representation that was derived from visual features. Semantic search systems can use such directions to interpret queries involving gradual and possibly ill-defined features, such as “*popular* holiday destinations in Europe” [12]. While features such as popularity are typically not encoded in traditional knowledge bases, they can often be represented as semantic space directions. As another application, directions can also be used in interpretable classifiers. For example, [6] learned rule based classifiers from ranks induced by the feature directions.

Other work which has taken advantage of directions in vector spaces has relied on word-embeddings 2.4.2. For instance, [9] found that features of countries, such as their GDP, fertility rate or even level of CO<sub>2</sub> emissions, can be predicted from word embeddings using a linear regression model. In [15] directional vectors in word embeddings were found that correspond to adjectival scales (e.g. bad < okay < good < excellent) while [27] found directions indicating lexical features such as the frequency of occurrence and polarity of words.

This work builds on the original method to find directions in a vector space and rank documents on them introduced by Derrac [6]. This chapter differs from that work by first introducing and explaining variants to the method in Section 4.2, then investigating directions qualitatively in Section ??, examining how the new variants perform relative to each other, and finally an extensive quantitative examination in section ?? of all variants across all domains (as described in 4.1) to determine their usefulness in interpretable classifiers is explored. Finally, conclusions are

made on the contribution of the chapter in section ?? . Chapter ?? builds on this method by applying and investigating its usage with vector spaces obtained from supervised neural networks, and the Chapter ?? identifies problems with this method and introduces a novel unsupervised solution to improve performance.

## 4.2 Method

This section details the methodology to add structure to a vector space model starting with only itself and its associated Bag-Of-Words ?? . The work in this chapter differs from the method introduced by Derrac [6] as it focuses on achieving a new representation that can be applied in simple interpretable classifiers. Multiple variations are introduced and experimented on comprehensively.

word:  $w$

document  $d$

vector-space of documents  $V_d$  where  $d = (x_1, x_2, \dots, x_n)$  where  $x$  are features and  $x(d)$  is equal to the value of a feature for a document

bag-of-words of documents  $B_d$  where  $d = (wf_1, wf_2, \dots, wf_n)$  and  $wf(d)$  is equal to the frequency of that word in a document and  $n$  is equal to the number of unique words across all documents.

model for a word ( $M_w$ )

hyper-plane for a word =  $H_w$

orthogonal direction vector =  $\mathcal{D}_w$

ranking of all documents on a word direction  $R_w = (rw_{d1}, rw_{d2}, \dots, rw_{dn})$  where  $rw_d$  is equal to the ranking of a document on a word direction

cluster of words  $C = (w_1, w_2, \dots, w_n)$

cluster direction =  $\mathcal{D}_C$

interpretable representation composed of rankings  $I_d$  where  $d = (R_{w1}, R_{w2}, \dots, R_{wn})$  and  $R_w(d)$  is equal to the ranking of a document on a word direction

interpretable representation composed of cluster rankings  $I_d$  where  $d = (R_{c1}, R_{c2}, \dots, R_{cn})$  and  $R_c(d)$  is equal to the ranking of a document on a cluster direction

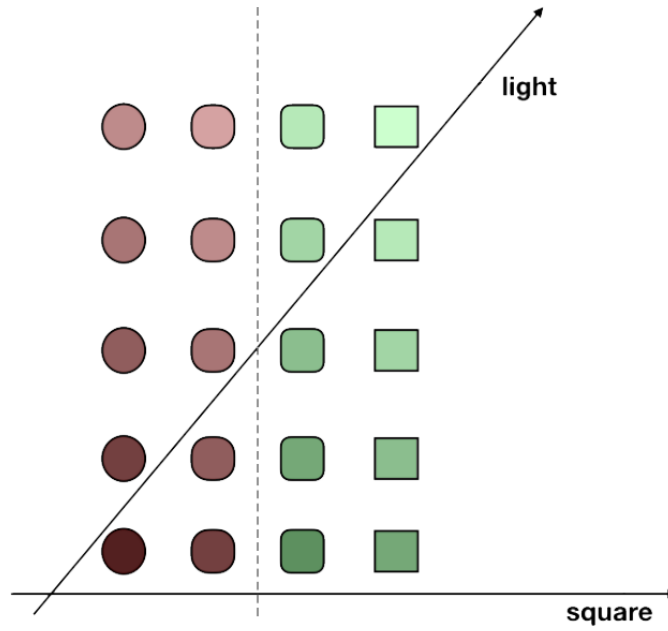
ranking of documents on direction

### 4.2.1 Obtaining Directions and Rankings From Words

The method starts with a given vector-space  $V_D$  induced from text documents  $d \in D$  and their associated bag-of-words  $B_D$ . For the bag-of-words  $B_D$  each document is composed of word frequencies  $d = (wf_1, wf_2, \dots, wf_n)$  where  $wf(d)$  is equal to the frequency of a word in a document and  $n$  is equal to the number of unique words in the vocabulary  $w \in W$ . Following the general explanation in the introduction, this section more precisely explains how to obtain a word-direction vector  $\mathcal{D}_w$  for all words in the vocabulary  $w \in W$ , by using a vector found by a linear model  $M_w$  that separates documents that have a word and do not have a word. Then, from that direction it explains how to obtain a ranking of all documents  $R_w = (rw_{d1}, rw_{d2}, \dots, rw_{dn})$  where  $rw_d$  is equal to the ranking of a document on a word direction and  $n$  is the number of documents. The section following this one shows how to remove word directions that are not semantically important by evaluating the quality of the classifier that obtained the direction  $M_w$ , or the quality of the direction  $\mathcal{D}_w$ .

**Obtaining directions for each word** Each document is represented by a vector  $v_d$  in the vector space model  $V_D$ . For this section, document vectors  $v_d$  are treated as points  $p_d$  in the space. For each word  $w$ , a hyper-plane  $h_w$  is obtained by training a linear model<sup>1</sup>  $M_w$  on the space  $V_D$  so that each document  $p_d$  in the space where the word  $w$  occurs more than once  $wf(d) \geq 1$  are separated from those where the word did not occur  $wf(d) = 0$ . We obtain such hyperplanes for all words in the vocabulary above a frequency threshold  $wf(D) > T$  where  $wf(D)$  is the frequency of the word in all documents. In practice, the parameter  $T$  is determined with hyperparameter optimization. This task is unbalanced, i.e. there are typically fewer documents that contain the word compared to those that do not contain it, so the weights of the classifier are

<sup>1</sup>Tested using a logistic regression classifier and a linear SVM, both achieved similar results



**Figure 4.2:** Another example of a hyper-plane in a toy domain of shapes. Here we show multiple directions, one for light and one for square. The hyper-plane for the word square is the dotted line. Green shapes are positive examples and red shapes are negative examples for the word square. .

balanced such that positive instances are weighted in proportion to how rare they are.<sup>2</sup>

Although the hyperplane  $h_w$  is classifying a binary class (either classifying documents  $d_p$  as negative or positive), the distance between the document vectors  $d_p$  and the hyperplane  $h_w$  will vary. For example, when separating documents based on the occurrence of a word, it can be expected that the documents which contain the word more frequently would be further away from the hyper-plane on the positive side. We give an example of two directions in Figure 4.2. To apply this idea to a real domain, we can give an example from movie reviews, where the word is 'Scary' and the most 'Scary' movies are at the tip of the direction and those that are least 'Scary' are at the base of the direction. With this understanding, the direction  $\mathcal{D}_w$  can be obtained by simply taking the vector perpendicular to the hyperplane  $h_w$ . This direction goes from documents  $d_p$  from those lowest on the direction (at the distance furthest from the hyperplane on the side where documents  $d_p$  are classified) to those highest on the direction at the distance furthest from the hyperplane at the positive side.

**Ranking documents on directions** In this section we specify how to obtain a ranking  $R_w$  of

<sup>2</sup>Using scikit-learn, class\_weight: 'balanced'



all documents on a word direction vector  $D_w$ . The rank of a document  $d$  can be defined by the dot product  $\mathcal{D}_w \cdot p_d$  as the ranking  $rw_d$  of the document  $d$  for the word  $w$ . Specifically,  $rw_{d_1}$  is ranked higher than  $rw_{d_2}$  if  $rw_{d_1} < rw_{d_2}$ . These rankings measure how relevant the document is in the spatial representation for the word, rather than just frequency e.g. a document that contains the word "scary" but isn't a scary movie (e.g. if it contained sentences like "it's scary how much money is spent on advertising movies like this") would not be ranked highly on the direction for 'scary', as the word 'scary' is not semantically important for the document. To put it another way, intuitively it can be understood to mean that the document  $d_2$  'has' the feature to a greater extent than  $d_1$ , e.g. in a domain of movie reviews if a movie ranked highly on the word 'dull', the movie has more dullness than lesser ranked movies.

In this section, the methodology to obtain word-directions and their associated rankings was described. These word-rankings are useful as features, and hypothetically we could obtain a representation that has as many features as there are words. However, some words are more semantically important in the domain than others. The next step describes how to remove word directions that are not well predicted by the linear model  $M_w$ , under the assumption that if they are not spatially important (i.e. easily separable), they are not semantically important. Another problem with word-rankings is that their meaning can be unclear, e.g. the word "serial" could be referring to a series of movies, or a "serial" killer<sup>3</sup>. To solve this problem, the final section explains methods to cluster words together. This finally results in a representation where each feature is semantically important and has an associated cluster of words to give context. We gave examples of these clusters of words in the introductory table 4.1.

### 4.2.2 Filtering Word-Directions

Although we are able to obtain word directions for every word, not every word is semantically important in the space. Here, we distinguish between word-directions  $\mathcal{D}_w$  as directions that are not semantically important in the space, and feature-directions  $\mathcal{D}_f$ , which are. We define the set of feature directions as  $f \in F$ . Additionally, we make the distinction between a word-ranking  $R_w$  which is a ranking on a word-direction and a feature-ranking which is a ranking on a feature-direction. This section how to filter out word-directions so that only feature-directions remain,

---

<sup>3</sup>The real cluster of words that this example comes from is "gore gory bloody blood gruesome serial investigate deaths"

in-order for the final representation to be composed of only feature-rankings. Additionally, we refer to the word or cluster of words associated with a feature-ranking as a feature-label for that ranking.

The assumption made by Derrac in the original work [6] was that if a classifier  $M_w$  does not predict the occurrence of a word in a document in its embedding well, it is not semantically important. Put another way, if the documents are separated well, it must mean that the word  $w$  being used in the description of  $d$  is important enough to affect the Vector Space Model representation of  $d$ . The occurrence of a word in a document can be evaluated by using a variety of scoring metrics to determine the performance of the classifier  $M_w$ . This work also introduces a the use of a scoring metric that evaluates the quality of the direction  $\mathcal{D}_w$ , as even if the documents are well separable, then the ranking induced from the direction may not be correct. This metric compares how well the ranking induced by the hyperplane correlates with a BoW representation. If the ranking does correlate, it can be assumed that this means the word was strongly influential in the space, as the detail of the Bag-Of-Words information is embedded in the space's structure.

After scoring the words using one of the aforementioned metrics, a simple cut-off is applied where the top scoring words are taken as feature-directions (e.g. the top 2000 scored words). By obtaining these feature-directions, we can rank documents on each of them and use them as features of a representation  $I_{Fr}$  where  $Fr = (fr_{d1}, fr_{d2}, \dots, fr_{dn})$  and  $fr_d$  is the ranking of a document on a feature-direction. In this representation, each feature is semantically important, however there may be overlap, e.g. the word "Gore" and "Gory" likely have similar rankings. Ideally, the score cut-off would be at the point where the words stop corresponding to semantically important features. However, it is difficult to determine this, so in practice this value is taken as a hyper-parameter determined by a classifier on some domain task.

**Cohen's Kappa.** This is the only metric used in the work by Derrac [6]. This metric evaluates the performance of the classifier, and also deals with the problem that these words are often very imbalanced. In particular, for very rare words, a high accuracy might not necessarily imply that the corresponding direction is accurate, as if there are a large number of negative examples (as is the case with infrequent words) the classifier could simply predict that all documents do not contain the word to achieve a high score. For this reason, they proposed to use Cohen's

Kappa score instead. In our experiments, however, it was found that this can be too restrictive, allowing us to sometimes obtain better results with the more simple accuracy metric.

**Classification accuracy.** If a model has high accuracy for a word  $w$ , it seems reasonable to assume that  $w$  describes a salient property for the given domain. However, despite balancing the weights of the original SVM used to obtain the hyper-plane, the value this metric places on correctly predicting negative classification compared to Kappa, it might favour rare words as it tends to be easier to obtain a high accuracy for these words.

**Normalized Discounted Cumulative Gain** This is the metric chosen to evaluate the quality of the rankings induced by the direction  $R_w$ . In-order to do so, two rankings are compared: one defined by the rankings and one defined by PPMI scores. The metric found to work best was Normalized Discounted Cumulative Gain (NDCG) which is a standard metric in information retrieval that evaluates the quality of a ranking w.r.t. some given relevance scores [14]. NDCG is mostly affected by the ranking position of the documents for which PPMI is highest. Spearman Rho, Gini, and Kendall Tau as alternative metrics do not favour higher ranked documents as much, but this comes with two problems. First, PPMI (See section ??) leads to a large number of zero scores. If we assume that all documents that have a zero frequency are ranked the same, then the dot products rankings will be greatly different for lower-ranked documents as they instead are ranked according to their spatial representation. This disrupts the score too much to be useful when lower ranked documents are given equal importance to higher ranked ones. In this case, the rankings  $R_w$  of the document  $d$  are those induced by the dot products  $\mathcal{D}_w \cdot p_d$ . The relevance scores are determined by the Pointwise Positive Mutual Information (PPMI) score  $PPMI(w, d)$ , of the word  $w$  in the BoW representation of document  $d$  (See section ??), under the assumption that they correspond to a good baseline for what we consider to be important for an entity.

$$\begin{aligned} DCG_R^w &= \sum_{i=1}^{pr_d} \frac{ppmi_i^w}{\log_2(i+1)} \\ IDCGR^w &= \sum_{i=1}^{|documents|} \frac{2^{ppmi_i^w} - 1}{\log_2(i+1)} \\ nDCGR^w &= \frac{DCGR^w}{IDCGR^w} \end{aligned}$$

To define NDCG, we can first define Discounted Cumulative Gain (DCG), where  $prw_d$  is equal

to a position in the ranking of documents on a direction  $\mathcal{D}_w$ , and  $ppmi_{p_i}^w$  is equal to the PPMI score for a word at position  $i$  in the ranking. Then, we can define the Ideal Discounted Cumulative Gain (IDCG), which is the best possible DCG for a position  $prw_d$ , where  $|documents|$  are the documents for the term ordered by their relevance up to position  $prw_d$ . nDCG is then simply the DCG normalized by the iDCG.

### 4.2.3 Clustering Features

A representation composed only of rankings of single words could be used, however that comes with two issues. The first is that there may be too many dimensions, so a classifier like a Decision Tree (See Section 2.3.2) needs to be deep to classify well. The second is that it can sometimes be ambiguous what a feature-ranking means when it is labelled with only a single word, e.g. the word "courage" is a feature-direction, but what it represents can only be understood in the context of its cluster "courage students teaches student schools teacher teach classes practice training learning overcome conflict teaching" showing that it is about courageous teachers and students overcoming challenges. There are two ways to solve this problem. The first is that the most similar directions (using cosine similarity) can be found and concatenated with the original word. However, this does not reduce the amount of features. By labelling feature-directions like this, we can rank documents on each of these directions to obtain associated labels for each feature that have context  $Fr_{wl}$  where  $wl = (wl_{fr1}, wl_{fr2}, \dots, wl_{frn})$  and  $wl_{fr}$  is a group of words to label a feature.

By obtaining these feature-directions, we can rank documents on each of them and use them as features of

The second is that the directions themselves can be clustered, such that feature-directions are clustered together and a new representation  $I_{Cr}$  is taken composed of rankings on feature cluster-directions  $Cr_i = (cr_{d1}, cr_{d2}, \dots, cr_{dn})$  where  $cr_d$  is the ranking of a document on a cluster feature-direction. Associated labels are obtained by simply concatenating the words of the feature-directions that are clustered together. These clustered feature-directions can be obtained for example by averaging all feature-directions that are clustered together. The benefits of this are that first, the number of features are reduced. Second, the words from the associated feature-directions that are clustered together can be used as feature-labels. Finally, two

directions that describe a similar feature of movies can be clustering together e.g. the feature-directions for the words "Bloody" and "Gorey". Both are words in movie reviews to describe how much blood a movie contains, so if these feature directions are averaged then the cluster direction can be used to produce a more balanced ranking for how much blood there is in films. Essentially, the cluster feature-direction could more accurately represent the semantics of a bloody film, compared to what is possible when considering either feature-direction individually.

On the other hand, its possible that when clustering many words together the cluster feature-direction no longer represents a semantically important feature. For example given the associated label for a cluster feature-direction  $\{Romance, Love\}$  and a cluster feature-direction  $\{Bloody, Gorey\}$  the feature-direction for  $\{Cute\}$  is more relevant to the former rather than the latter, and has been used in reviews for romance movies. But it has also been used in reviews for movies containing cute animals. This would make the new clustered direction  $\{Romance, Love, Cute\}$  perform worse at classifying the movie genre "Romance", but a bit better at classifying if a movie contains animals. It might thus be preferable to keep *Cute* in a separate cluster for animal movies - but a balance must be struck between finding the most semantically important clusters in the space and creating new clusters that may not be as semantically important because that word does not easily fit into a cluster. In the quantitative results, sometimes clustering performed worse than single directions, and not being able to find this balance for the specific classes in question can be attributed as to why, specifically because clusters were not semantically important enough or were disrupted by clustering together words that do not fit.

We will experiment with two different clustering methods: k-means and a variant of k-means that was proposed in the work by Derrac [6]:

**K-Means** In the experimental results, it was found that Derrac's variation relies too much on its initial directions, meaning if a noisy direction is chosen as the first cluster centre, then key directions may be missed. Avoiding this is difficult without extensive and sometimes arbitrary hyper-parameter optimization. For this reason, it was decided to also consider K-Means as an alternative clustering algorithm. K-means traditionally begins with  $K$  centroids  $c$  randomly placed into the space. In our case, these centers are weighted according to the squared distance from the closest center already chosen. [1] Then, the distance between each point  $d_p$  and

centroid  $c$  is calculated. In-order for Euclidean distance to be meaningful, directions are normalized making Euclidean distance the same as cosine similarity. Each point  $p$  is then assigned to its closest centroid  $c$ . Then, the centroids are recomputed to be the mean of their assigned points. This process starting with the distance calculation is repeated until the points assigned to the centroids do not change.

**Derrac’s K-Means Variation** This is the clustering method used in the work this method was introduced in [?]. As input to the clustering algorithm, it considers the  $N$  best-scoring candidate feature directions  $v_w$ , where  $N$  is a hyperparameter. The main idea underlying their approach is to select the cluster centers such that (i) they are among the top-scoring candidate feature directions, and (ii) are as close to being orthogonal to each other as possible.

The output of this step is a set of clusters  $C_1, \dots, C_K$ , where each cluster  $C_j$  is identified with a set of words. In the following, we will write  $v_{C_j}$  will be written to denote the centroid of the directions corresponding to the words in the cluster  $C_j$ , which can be computed as  $v_{C_j} = \frac{1}{|C_j|} \sum_{w_l \in C_j} \frac{v_l}{\|v_l\|}$  provided that the vectors  $v_w$  are all normalized.

The first cluster centroid is chosen by taking the top-scoring direction for a scoring metric. Then, centroids are selected until the desired number is reached by taking the maximum of the summed absolute cosine similarity of all current centroids, in other words taking the most dissimilar direction to all of the current directions. Once the centroids are selected, for each remaining direction the centroid is found it is most similar to, and the centroid is updated once the direction has been added.

Meaning that the key is to rank documents on the initial direction only, and only use the remaining features in each cluster to provide a more informative label if the clusters are too noisy.

## 4.3 Qualitative Results

In principle, NDCG should be better suited for gradual features. For example, a binary feature would be 'Gore', where a film is either gory or not gory. A gradual feature would be "rating", referring to the age rating for films and gradually increasing. In practice, however, there was not such a clear pattern in the differences between the words chosen by these metrics despite

often finding different words. Put another way, it is difficult to say if the words highly scored by NDCG are more gradual than other scoring metrics.

### 4.3.1 Datasets

For each domain, we filter out terms that do not occur in at least two documents, and additionally limit the maximum number of words in a vocabulary to 100,000. For all of these datasets, we split them into a 2/3 training data, 1/3 test data split. We additionally remove the end 20% of the training data and use that as development data for our hyper-parameters, which is then not used for the final models verified using test data. For the movies and place-type domains, the original text was not available.

### 4.3.2 Space Types

Below the choices for the Vector Space Models that are formally described in Section ?? are explained:

**Multi-Dimensional Scaling (MDS):** Following [5], we use Multi-Dimensional Scaling (MDS) to learn semantic spaces from the angular differences between the PPMI weighted BoW vectors.

**Principal Component Analysis (PCA):** directly uses the PPMI weighted BoW vectors as input, and which avoids the quadratic complexity of the MDS method. A standard dimensionality reduction technique, used as a baseline reference.

**Doc2Vec (D2V):** Inspired by the Skipgram model [18]. A distributional document representation used as a representative of a higher performing method of learning in terms of document classification. For the Doc2Vec space, the hyper-parameters are additionally tuned for the *window size*(5, 10, 15) referring to the context window, the *mincount*(1, 5, 10) referring to the minimum frequency of words and the *epochs*(50, 100, 200) of the network for each size space. The process with our two-part hyperparameter optimization as in this case is as follows: Grid search is used to select the parameters for the representation, then find the most suitable model (e.g. Decision Tree, SVM) for that representation.

**Average Word Vectors (AWV):** Finally, we also learn semantic spaces by averaging word vectors, using a pre-trained GloVe word embeddings trained on the Wikipedia 2014 + Gigaword 5 corpus<sup>4</sup>. While simply averaging word vectors may seem naive, this was found to be a competitive approach for unsupervised representations in several applications [10]. We simply average the vector representations of the words that appear at least twice in the BoW representation.

### 4.3.3 The best-performing directions for each domain

To give an understanding of the kind-of directions found for each domain, the top-scoring ones are presented in Table 4.2. These are arranged from highest scoring to least scoring, with the score-type and space-type chosen by performance. These are not clusters, but rather single directions with the two most similar directions in brackets beside them for context. This is the alternative way of presenting these directions as mentioned at the start of Section 4.2.3.

There is an interesting difference between the sentiment directions and the movies directions in the examples below. Both of these domains are composed of movie reviews, but the documents in the former are a concatenation of a number of reviews across different sources, while the latter are individual reviews. This has resulted in the more general concepts that apply to many movies being salient in the movies domain, but are less important than the names of actors and actresses in the sentiment domain. This is likely because the PPMI scores for actor names would be high as they are both rare and definitive for movies. For the newsgroups domain, a number of directions are seen that are likely to only belong to a certain newsgroups, e.g. you would find the word 'celestial' more often in the religious sections than the others, and the word 'diesel' more often in the automobile section but not others. This is an expected natural clustering of the domain into its 20 newsgroups. The place-types section generally describes either aspects of the camera (e.g. canon60d), aspects of the photo (greyscale) or features found in the photo (gardening). The former likely relates to the degree to which filters or editing has been applied to the photo, while the latter makes more sense for our classification task. For the reuters dataset, the highest scored semantics seem to generally be related to dates (1st, may, june), however there is also some business jargon (quarterly, avg, dlr).

---

<sup>4</sup><https://nlp.stanford.edu/projects/glove/>



| <b>Movies</b> (50 MDS NDCG)          | <b>Sentiment</b> (100 D2V NDCG) | <b>News</b> (50 D2V NDCG)            | <b>Place-types</b> (50 PCA Kappa)       | <b>Reuters</b> (200 MDS NDCG)     |
|--------------------------------------|---------------------------------|--------------------------------------|---|-----------------------------------|
| horror (scares, scary)               | glenda (glen, matthau)          | karabag (iranian, turkiye)           | blackcountry (listed, westmidlands)     | franklin (fund, mthly)            |
| hilarious (funniest, hilarity)       | scarlett (gable, dalton)        | leftover (flaming, vancouver)        | ears (stare, adorable)                  | quarterly (shearson, basis)       |
| bollywood (hindi, india)             | giallo (argento, fulci)         | wk (5173552178, 18084tmibmchmsuedu)  | spagna (espanha, colores)               | feb (28, splits)                  |
| laughs (funnier, funniest)           | bourne (damon, cusack)          | 1069 (mlud, wibbled)                 | oldfashioned (winery, antiques)         | 22 (booked, hong)                 |
| jokes (gags, laughs)                 | piper (omen, knightley)         | providence (norris, ahl)             | gardenng (greenhouse, petals)           | april (monthly, average)          |
| comedies (comedic, laughs)           | casper (dolph, damme)           | celestial (interplanetary, bible)    | pagoda (hindu, carved)                  | sets (principally, precious)      |
| hindi (bollywood, india)             | norris (chuck, rangers)         | mlud (wibbled, 1069)                 | artificial (saturation, cs4)            | 16 (creditor, trillion)           |
| war (military, army)                 | holmes (sherlock, rathbone)     | endif (olwm, ciphertxt)              | inner (curved, rooftops)                | 1st (qtr, pennsylvania)           |
| western (outlaw, unforgiven)         | rouke (mickey, walken)          | gd3004 (35894, intergraph)           | celebrate (festive, celebrity)          | 26 (approve, inadequate)          |
| romantic (romance, chemistry)        | ustinov (warden, cassavetes)    | rftmitedu (newsanswers, ieee)        | vietnamese (ethnic, hindu)              | 23 (offsetting, weekly)           |
| songs (song, tunes)                  | scooby (doo, garfield)          | eng (padres, makefile)               | cn (elevated, antrak)                   | prior (recapitalization, payment) |
| sci (science, outer)                 | doo (scooby, garfield)          | pizza (bait, wiretap)                | mannequin (bags, jewelry)               | avg (shrs, shr)                   |
| funniest (hilarious, funnier)        | heston (charlton, palance)      | porsche (nanao, mercedes)            | falcon (r, 22)                          | june (july, venice)               |
| noir (noirs, bogart)                 | homer (pacino, macy)            | gebeadredspitedu (n3jxp, skepticism) | jewish (monuments, cobblestone)         | march (31, day)                   |
| documentary (documentaries, footage) | welles (orson, kane)            | scsi2 (scsi, cooling)                | canon60d (kitlens, 600d)                | regular (diesel, petrol)          |
| animation (animated, animators)      | frost (snowman, damme)          | playback (quicktime, xmotif)         | reflective (curved, cropped)            | 4th (qtr, fourth)                 |
| adults (adult, children)             | streisand (bridget, salman)     | 35894 (gd3004, medin)                | mason (edward, will)                    | 27 (chemlawn, theyre)             |
| creepy (spooky, scary)               | davies (rhys, marion)           | diesel (volvo, shotguns)             | aerialview (manmade, largest)           | 14 (borrowing, borrowings)        |
| gay (gays, homosexuality)            | cinderella (fairy, stepmother)  | evolutionary (shifting, hulk)        | shelf (rack, boxes)                     | 11 (chapter, ranged)              |
| workout (intermediate, instruction)  | boll (uwe, belushi)             | techniciandr (obp, 144k)             | monroe (raleigh, jefferson)             | may (probably, however)           |
| thriller (thrillers, suspense)       | rochester (eyre, dalton)        | 8177 (obp, 144k)                     | litter (fujichrome, e6)                 | 38 (33, strong)                   |
| funnier (laughs, funniest)           | edie (soprano, vertigo)         | shaw (medicine, ottoman)             | streetlights (streetlamp, headlights)   | m1 (m2, m3)                       |
| suspense (suspenseful, thrillers)    | scarecrow (zombies, reese)      | scorer (gilmour, lindros)            | carlzeiss (f2, voigtlander)             | dlr (writedown, debt)             |
| arts (hong, chan)                    | kramer (strep, meryl)           | xwd (xloadimage, openwindows)        | manmade (aerialview, below)             | five (years, jones)               |
| christianity (religious, religion)   | marty (amitabh, goldie)         | ee (275, xloadimage)                 | demolished (neglected, rundown)         | bushels (soybeans, ccc)           |
| musical (singing, sing)              | columbo (falk, garfield)        | com2 (com1, v32bis)                  | wald (berge, wildflower)                | revs (net, 3for2)                 |
| gore (gory, blood)                   | kidman (nicole, jude)           | examiner (corpses, brass)            | arquitectura (exposition, cidade)       | 29 (175, include)                 |
| animated (animation, cartoon)        | juliet (romeo, troma)           | migraine (ama, placebo)              | greyscale (highcontrast, monochromatic) | acquisition (make, usairs)        |
| gags (jokes, slapstick)              | garland (judy, lily)            | parliament (parliamentary, armored)  | alameda (monday, marin)                 | payable (div, close)              |

Table 4.2: The top-scoring words for each domain, scoring metric and space type determined by the highest F1-score

### 4.3.4 Comparing Space Types

To select these quantitative examples for comparing score types, it was first demonstrated on the movies domain to be consistent with previous examples. However, as this does not contain the doc2vec space, additional results are provided in the next section for the newsgroups. The space that performed well on the genres task for the movies is used, with the understanding that genres as a key natural classification task will likely give good example directions that correspond to domain knowledge. After selecting this space, the same sized spaces are chosen from the other space-types (size 200). The same score-type and frequency cut-off as the best performing space-type are also used. In this case, the best performing type for the PCA space was 20,000 frequency cutoff and NDCG. So even though sometimes a different frequency cut-off performed better for the other space-types, this is equalized so that the words are the same. This means that sometimes the space-type is a slightly worse performing one than chosen as the final results, and that the original space has a performance advantage, but this makes the results more consistent. These qualitative experiments are approached with the following idea: spaces that perform better on natural domain tasks using Decision Trees contain unique natural directions that other spaces do not have.

The commonalities between spaces are much more prevalent than the differences, with natural concepts of the domain being represented in all of the different space types. However, different spaces do perform better than others on natural domain tasks. For this reason, the directions which are unique to each space-type are shown.

When examining the table of results, it can be observed that the common terms are mostly salient concepts relevant to the domain. However, MDS has the most unique general concepts relevant to the domain that others do not have. AWV contains names, and concepts which are interesting but more related to specific aspects than genre (train, slaves). Meanwhile PCA seems to prioritize words in the reviews that are not concepts but rather parts of sentences (surprisingly, admit, talents, tired, anymore). However, both PCA and MDS contain unique noisy terms as well. The term 'berardinelli' and 'rhodes' for MDS as well as 'compuserve' for PCA are artifacts of the data being obtained from the web. Despite this, it seems that MDS does contain more interesting unique directions than PCA, and as it performed best on the genres task, this makes sense.

| MDS  | AWV                               | PCA  | Common   |
|--|-----------------------------------|--|--|
| berardinelli ( <i>employers, distributor</i> ) | billy ( <i>thrown, dirty</i> )    | amount ( <i>leaving, pick</i> )                  | noir ( <i>fatal, femme</i> )                     |
| crawford ( <i>joan, davis</i> )                | brother ( <i>brothers, boys</i> ) | fails ( <i>fit, pick</i> )                       | gay ( <i>homosexual, homosexuality</i> )         |
| hitchcocks ( <i>hitchcock, alfred</i> )        | fonda ( <i>henry, jane</i> )      | pick ( <i>fails, fit</i> )                       | prison ( <i>jail, prisoners</i> )                |
| warners ( <i>warners, bros</i> )               | building ( <i>built, climax</i> ) | stands ( <i>fails, cover</i> )                   | arts ( <i>rec, robomod</i> )                     |
| nuclear ( <i>weapons, soviet</i> )             | train ( <i>tracks, thrown</i> )   | surprisingly ( <i>offer, fit</i> )               | allens ( <i>woody, allen</i> )                   |
| joan ( <i>crawford, barbara</i> )              | slaves ( <i>slavery, excuse</i> ) | copyright ( <i>email, compuserve</i> )           | jokes ( <i>laughs, joke</i> )                    |
| kidnapped ( <i>kidnapping, torture</i> )       |                                   | length ( <i>reflect, expressed</i> )             | animation ( <i>animated, cartoon</i> )           |
| hop ( <i>hip, rap</i> )                        |                                   | profanity ( <i>reflect, producers</i> )          | sherlock ( <i>holmes, detective</i> )            |
| kung ( <i>martial, jackie</i> )                |                                   | compuserve ( <i>copyright, internetreviews</i> ) | western ( <i>westerns, wayne</i> )               |
| ballet ( <i>dancers, dancer</i> )              |                                   | talents ( <i>admit, agree</i> )                  | songs ( <i>song, lyrics</i> )                    |
| gambling ( <i>vegas, las</i> )                 |                                   | admit ( <i>agree, talents</i> )                  | comedies ( <i>comedy, laughs</i> )               |
| alcoholic ( <i>drunk, alcoholism</i> )         |                                   | developed ( <i>introduced, sounds</i> )          | workout ( <i>exercise, challenging</i> )         |
| waves ( <i>surfing, wave</i> )                 |                                   | intended ( <i>bother, weren't</i> )              | laughs ( <i>funnier, hilarious</i> )             |
| jaws ( <i>jurassic, godfather</i> )            |                                   | constantly ( <i>putting, sounds</i> )            | drug ( <i>drugs, addict</i> )                    |
| jungle ( <i>natives, island</i> )              |                                   | tired ( <i>anyone, mediocre</i> )                | sci ( <i>science, fiction</i> )                  |
| employers ( <i>berardinelli, distributor</i> ) |                                   | produced ( <i>spoiler, surprising</i> )          | documentary ( <i>documentaries, interviews</i> ) |
| pot ( <i>weed, stoned</i> )                    |                                   | involving ( <i>believes, belief</i> )            | students ( <i>student, schools</i> )             |
| canadian ( <i>invasion, cheap</i> )            |                                   | anymore ( <i>continue, tired</i> )               | thriller ( <i>thrillers, suspense</i> )          |
| murphy ( <i>eddie, comedian</i> )              |                                   | leaving ( <i>fit, pick</i> )                     | allen ( <i>woody, allens</i> )                   |
| comics ( <i>comedian, comedians</i> )          |                                   | makers ( <i>producers, aspects</i> )             | funniest ( <i>hilarious, laughing</i> )          |
| kidnapping ( <i>kidnapped, torture</i> )       |                                   | introduced ( <i>developed, considered</i> )      | gags ( <i>jokes, slapstick</i> )                 |
| subscribe ( <i>email, internetreviews</i> )    |                                   | loses ( <i>climax, suffers</i> )                 | adults ( <i>children, adult</i> )                |
| vegas ( <i>las, gambling</i> )                 |                                   | negative ( <i>positive, bother</i> )             | animated ( <i>animation, cartoon</i> )           |
| distributor ( <i>berardinelli, employers</i> ) |                                   | expressed ( <i>reflect, opinions</i> )           | dancing ( <i>dance, dances</i> )                 |
| wave ( <i>waves, surfing</i> )                 |                                   | mildly ( <i>mediocre, forgettable</i> )          | teen ( <i>teenage, teens</i> )                   |
| rhodes ( <i>internetreviews, email</i> )       |                                   | helped ( <i>putting, allowed</i> )               | soldiers ( <i>soldier, army</i> )                |
| hippie ( <i>pot, sixties</i> )                 |                                   | reflect ( <i>expressed, opinions</i> )           | indie ( <i>independent, festival</i> )           |
| weed ( <i>pot, stoned</i> )                    |                                   | opinions ( <i>reflect, expressed</i> )           | suspense ( <i>suspenseful, thriller</i> )        |
| caribbean ( <i>pirates, island</i> )           |                                   | frequently ( <i>occasionally, consistently</i> ) | creepy ( <i>scary, eerie</i> )                   |
| eddie ( <i>murphy, comedian</i> )              |                                   | content ( <i>agree, proves</i> )                 | italian ( <i>italy, spaghetti</i> )              |
| sixties ( <i>beats, hippie</i> )               |                                   | allowed ( <i>helped, weren't</i> )               | jews ( <i>jewish, nazis</i> )                    |
| ... 8 More                                     |                                   | suffers ( <i>lacks, loses</i> )                  | ... 1480 more                                    |

Table 4.3: Unique terms between space-types

## Score Types

There are unique directions for each different space type from the movies domain, each suitable to different tasks. Obtained in the same way as before, this time the 200 MDS space is used that performed the best on the genres task and found those unique to it. Once again, the most understandable and general concepts are those that are common to all score-types. NDCG performed the best on most tasks, and it can be seen that a lot of new concepts are introduced in NDCG compared to the other scoring types. F1 by and large seems is difficult to understand, referring to names or specific aspects of the scene, and accuracy is similar. Kappa has some unique sentiment related terms, as well as some aspects of the presentation of the film (featurette, critic, technical), but it does not contain unique general concepts the way NDCG does. It can be surmised that as NDCG contains these unique conceptual directions, it is able to perform better than other score-types.

| NDCG                                       | F1  | Accuracy   | Kappa                                       | Common  |
|--|---|--|---|---|
| gay ( <i>homosexuality, sexuality</i> )    | company ( <i>sell, pay</i> )              | kennedy ( <i>republic, elected</i> )             | definitely ( <i>alot, awesome</i> )         | horror ( <i>scares, scares</i> )              |
| arts ( <i>hong, chan</i> )                 | street ( <i>city, york</i> )              | bags ( <i>listened, salvation</i> )              | guns ( <i>gun, shoot</i> )                  | laughs ( <i>funnier, funnier</i> )            |
| sports ( <i>win, players</i> )             | red ( <i>numerous, fashion</i> )          | summers ( <i>verge, medieval</i> )               | flawless ( <i>perfection, brilliantly</i> ) | jokes ( <i>gags, gags</i> )                   |
| apes ( <i>remembered, planet</i> )         | project ( <i>creating, spent</i> )        | revolve ( <i>sincerely, historian</i> )          | mail ( <i>reviewed, rated</i> )             | comedies ( <i>comedic, comedic</i> )          |
| german ( <i>germans, europe</i> )          | mark ( <i>favor, pull</i> )               | locale ( <i>foster, sharply</i> )                | garbage ( <i>crap, horrible</i> )           | sci ( <i>scifi, alien</i> )                   |
| satire ( <i>parody, parodies</i> )         | lady ( <i>actress, lovely</i> )           | cooler ( <i>downward, reports</i> )              | featurette ( <i>featurettes, extras</i> )   | funniest ( <i>hilarious, hilarious</i> )      |
| band ( <i>rock, vocals</i> )               | fire ( <i>ground, force</i> )             | spades ( <i>ralph, medieval</i> )                | complaint ( <i>extra, added</i> )           | creepy ( <i>spooky, spooky</i> )              |
| crude ( <i>offensive, offended</i> )       | post ( <i>essentially, purpose</i> )      | filmography ( <i>ralph, experiments</i> )        | mission ( <i>enemy, saving</i> )            | thriller ( <i>thrillers, thrillers</i> )      |
| dancing ( <i>dance, dances</i> )           | heads ( <i>large, throw</i> )             | quentin ( <i>downward, anime</i> )               | ruin ( <i>wondering, heck</i> )             | funnier ( <i>laughs, laughs</i> )             |
| restored ( <i>print, remastered</i> )      | water ( <i>land, large</i> )              | employers ( <i>finishes, downward</i> )          | wars ( <i>forces, enemy</i> )               | suspense ( <i>suspenseful, suspenseful</i> )  |
| drugs ( <i>drug, abuse</i> )               | road ( <i>drive, trip</i> )               | formal ( <i>victory, kennedy</i> )               | prefer ( <i>compare, added</i> )            | gore ( <i>gory, gory</i> )                    |
| church ( <i>religious, jesus</i> )         | brother ( <i>son, dad</i> )               | tube ( <i>esta, muscle</i> )                     | heroes ( <i>packed, hero</i> )              | gags ( <i>jokes, jokes</i> )                  |
| sexuality ( <i>sexual, sexually</i> )      | party ( <i>decide, hot</i> )              | woefully ( <i>restless, knockout</i> )           | necessarily ( <i>offer, draw</i> )          | science ( <i>sci, sci</i> )                   |
| sexually ( <i>sexual, sexuality</i> )      | badly ( <i>awful, poorly</i> )            | scientists ( <i>hilarity, locale</i> )           | portray ( <i>portrayed, portraying</i> )    | gory ( <i>gore, gore</i> )                    |
| england ( <i>british, english</i> )        | limited ( <i>aspect, unlike</i> )         | overboard ( <i>civilized, chiderella</i> )       | critic ( <i>reviewed, net</i> )             | government ( <i>political, political</i> )    |
| ocean ( <i>sea, boat</i> )                 | impression ( <i>instance, reasons</i> )   | rumors ( <i>homosexuality, characteristics</i> ) | reviewed ( <i>rated, mail</i> )             | suspenseful ( <i>suspense, suspense</i> )     |
| marry ( <i>married, marriage</i> )         | trip ( <i>journey, road</i> )             | salvation ( <i>bags, cooler</i> )                | saving ( <i>carry, forced</i> )             | frightening ( <i>terrifying, terrifying</i> ) |
| campy ( <i>cult, cheesy</i> )              | michael ( <i>producers, david</i> )       | actively ( <i>assassination, overcoming</i> )    | technical ( <i>digital, presentation</i> )  | military ( <i>army, army</i> )                |
| christian ( <i>religious, jesus</i> )      | memory ( <i>forgotten, memories</i> )     | stretching ( <i>victory, hideous</i> )           | statement ( <i>exist, critical</i> )        | slapstick ( <i>gags, gags</i> )               |
| melodrama ( <i>dramatic, tragedy</i> )     | james ( <i>robert, michael</i> )          | downward ( <i>cooler, crawling</i> )             | shocked ( <i>hate, warning</i> )            | scary ( <i>scare, scare</i> )                 |
| sing ( <i>singing, sings</i> )             | thin ( <i>barely, flat</i> )              | rocked ( <i>staple, demented</i> )               | flying ( <i>air, force</i> )                | blu ( <i>unanswered, ray</i> )                |
| sentimental ( <i>touching, sappy</i> )     | pre ( <i>popular, include</i> )           | affectionate ( <i>esta, muscle</i> )             | danger ( <i>dangerous, edge</i> )           | internetreviews ( <i>rhodes, rhodes</i> )     |
| depressing ( <i>bleak, suffering</i> )     | faces ( <i>constant, unlike</i> )         | protest ( <i>protective, assassination</i> )     |   | cgi ( <i>computer, computer</i> )             |
| evidence ( <i>investigation, accused</i> ) | values ( <i>exception, wise</i> )         | confined ( <i>cooler, downward</i> )             |   | email ( <i>web, web</i> )                     |
| adorable ( <i>cute, sweet</i> )            | unusual ( <i>odd, seemingly</i> )         | inhabit ( <i>quentin, drawback</i> )             |   | thrilling ( <i>thrill, exciting</i> )         |
| episodes ( <i>episode, television</i> )    | lovers ( <i>lover, lovely</i> )           | latin ( <i>communities, mount</i> )              |   | web ( <i>email, email</i> )                   |
| teenager ( <i>teen, teenage</i> )          | frame ( <i>image, effect</i> )            | reception ( <i>como, finishes</i> )              |   | horror ( <i>scares, scares</i> )              |
| magical ( <i>fantasy, lovely</i> )         | mans ( <i>ultimate, sees</i> )            | uptight ( <i>suspensful, stalked</i> )           |   | laughs ( <i>funnier, funnier</i> )            |
| health ( <i>medical, suffering</i> )       | efforts ( <i>generally, nonetheless</i> ) | brink ( <i>inexplicable, freddy</i> )            |   | suspense ( <i>suspenseful, suspenseful</i> )  |

Table 4.4: Different score types

### Comparing PPMI representations to doc2vec

Now in Table a comparison is shown between a time when doc2vec was the highest performing representation, in this case on the newsgroups domain. Doc2vec is compared to MDS in this case as MDS also performed well. This is to see if doc2vec, by making use of word-vectors and word-context can find interesting unique directions compared to MDS, which was obtained from a PPMI BOW. In general, it is found that MDS contains a lot more irrelevant words than D2V, specifically related to parts-of-words. It seems that doc2vec was better at recognizing these words as noise and uninteresting compared to PPMI, which must have prioritized these words. Doc2Vec also brings up some interesting concepts, e.g. cryptology, which is very relevant to the 20 newsgroup subtype of cryptography. It can be expected that by using word vectors, doc2vec is able to more easily identify interesting words and de-prioritize words which are common to the english language despite potentially being more rare in a smaller dataset.

## 4.4 Quantitative Results

For all of these datasets, we split them into a 2/3 training data, 1/3 test data split. We additionally remove the end 20% of the training data and use that as development data for our hyper-parameters, which is then not used for the final models verified using test data.

### 4.4.1 Evaluation Method

Primarily the effectiveness of a representation is evaluated on its ability to perform in low-depth Decision Trees, specifically CART Decision Trees (See Background Section 2.3.2) with a limited depth of one, two and three. This evaluation has a few assumptions: A good interpretable representation disentangles salient domain knowledge into its dimensions, and natural domain tasks (e.g. classifying genres of movies using their reviews) can be evaluated effectively using that salient domain knowledge. Put another way, if the space is representing domain knowledge well it can be expected that the space is linearly separable for key semantics of the domain. In spatial terms, a representation will be capable of being linearly transformed by our method

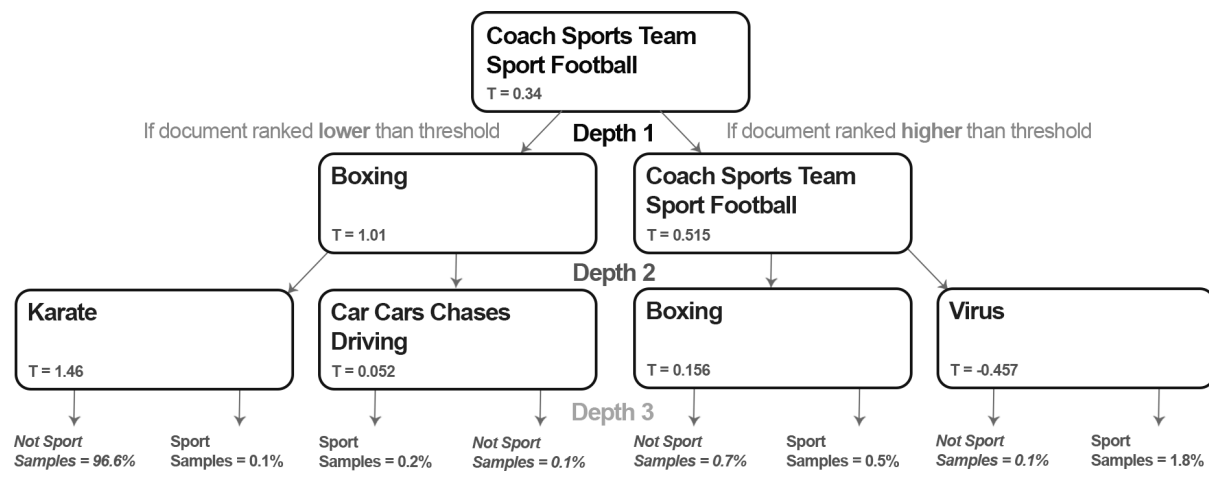
| D2V                                    | MDS                              | Common                                     |
|--|----------------------------------|--|
| leftover (pizza, brake)                | hi (folks, everyone)             | chastity (shameful, soon)                  |
| wk (5173552178, 18084tmibmclmsuedu)    | looking (spend, rather)          | n3jxp (gordon, gebcadredslpittedu)         |
| eng (padres, makefile)                 | need (needs, means)              | skepticism (gebcadredslpittedu, n3jxp)     |
| porsche (nanao, 1280x1024)             | post (summary, net)              | anyone (knows, else)                       |
| diesel (cylinders, steam)              | find (couldnt, look)             | gebcadredslpittedu (soon, gordon)          |
| scorer (gilmour, lindros)              | hello (kind, thank)              | intellect (soon, gordon)                   |
| parliament (caucasus, semifinals)      | david (yet, man)                 | please (respond, reply)                    |
| atm (padres, inflatable)               | got (mine, youve)                | thanks (responses, advance)                |
| cryptology (attendees, bait)           | go (take, lets)                  | email (via, address)                       |
| intake (calcium, mellon)               | question (answer, answered)      | know (let, far)                            |
| 433 (366, 313)                         | interested (including, products) | get (wait, trying)                         |
| ghetto (warsaw, gaza)                  | list (mailing, send)             | think (important, level)                   |
| lens (lenses, ankara)                  | sorry (guess, hear)              | good (luck, bad)                           |
| rushdie (sinless, wiretaps)            | heard (ever, anything)           | shafer (dryden, nasa)                      |
| immaculate (porsche, alice)            | cheers (kent, instead)           | bobbeviceicotekcom (manhattan, beauchaine) |
| keenan (lindros, bosnian)              | say (nothing, anything)          | dryden (shafer, nasa)                      |
| boxer (jets, hawks)                    | number (call, numbers)           | im (sure, working)                         |
| linden (mogilny, 176)                  | mailing (list, send)             | sank (bronx, away)                         |
| candida (yeast, noing)                 | call (number, phone)             | banks (soon, gordon)                       |
| octopus (web, 347)                     | thank (thanx, better)            | like (sounds, looks)                       |
| czech (detectors, kuwait)              | read (reading, group)            | shameful (soon, gordon)                    |
| survivor (warsaw, croats)              | phone (company, number)          | could (away, bobbeviceicotekcom)           |
| 5173552178 (circumference, wk)         | mail (send, list)                | would (appreciate, wouldnt)                |
| 18084tmibmclmsuedu (circumference, wk) | doesnt (isnt, mean)              | beauchaine (bobbeviceicotekcom, away)      |
| 3369591 (circumference, wk)            | lot (big, little)                | ive (seen, never)                          |
| mcwilliams (circumference, wk)         | thats (unless, youre)            | surrender (soon, gebcadredslpittedu)       |
| coldblooded (dictatorship, czech)      | believe (actually, truth)        | problem (problems, fix)                    |
| militia (federalist, occupying)        | youre (unless, theyre)           | windows (31, dos)                          |
| cbc (ahl, somalia)                     | send (mail, mailing)             | gordon (soon, gebcadredslpittedu)          |

**Table 4.5: Comparing an MDS sapce to a D2V space for Newsgroups, where a D2V space performed best..**

4.3.4

into these distinct relevant concepts if semantically distinct entities are spatially separated, and semantically similar entities are close together.

If only the the quality of the representation was being evaluated, only Linear SVM's could be used to find the hyper-planes that effectively separate these spatial representations for the class. However, the representations that encode this spatial information are not interpretable, so a linear classifier although able to separate the documents that contain the class and do not contain them will not be interpretable either. It is our main interest to evaluate how well a representation encodes these key semantics while also being restricted by the requirement to be



**Figure 4.3:** An example of a Decision Tree classifying if a movie is in the "Sports" genre. Each Decision Tree Node corresponds to a feature, and the threshold  $T$  is the required ranking of a document on that feature to traverse right down the tree instead of left. One interesting point to note is that the most important direction is used twice, the "coach, sports, team, sport, football" cluster and results in a majority of negative samples. Another point is that the nodes at depth three are more specific, sometimes overfitting (e.g. in the case of the "Virus" node, likely overfitting to a single movie about a virus) .

disentangled into words or clusters, in other words how well it represents the information while also being interpretable.

Given these assumptions, low-depth Decision Trees can give an estimation of how good an interpretable representation is. If the representation cannot perform for a class at a one-depth tree, then it is not disentangled such that it contains a single salient dimension that effectively evaluates a class. If a representation cannot perform well on two-depth trees, then the representation is not disentangled into three concepts that can sufficiently determine that class, and if a representation cannot perform well on three-depth trees, it has not disentangled the representation such that there are nine relevant concepts that are relevant to that class. To see what these different trees look like see Figure ?? . A comparison to put this in better perspective is to an unbounded tree. Unbounded trees select a large amount of dimensions in order to achieve a performance difference on development data, but when applied to test data the models do not generalize well. This is because they overfit, rather than using the key semantics of the space to classify.



Primarily F1-score is used to determine if a classifier is good or not. This is because many of the classes are unbalanced so accuracy is not a good metric, as high accuracy could be achieved by predicting only zeros. All of the results shown in this section are the end-product of a two-part hyper-parameter optimization. Each Decision Tree has its own set of hyper-parameters that are optimized as does each representation-type. These are the models trained on the training data and scored on the test data, with the highest performing in terms of F1-score parameters from hyper-parameter optimization on the development data. For ease of comparison, some results are provided with SVM's and unbounded Decision Trees, as well as a baseline Topic Model, which is used as a reference for a standard interpretable representation. Below, the parameters are listed that are optimized for each of these model types:

**Linear Support Vector Machines (SVM's)**2.5: C parameters and gamma parameters. C 1.0, 0.01, 0.001, 0.0001, Gamma 1.0, 0.01, 0.001, 0.0001.

**Topic Models**2.8: Two priors: The doc topic prior 0.001, 0.01, 0.1 and the topic word prior 0.001, 0.01, 0.1

**CART Decision Trees** 2.3.2: The number of features to consider when looking for the best split. *None, auto, log2* and the criterion for a node split *criterion : gini, entropy*.

For the baselines, four different Vector Space Models are used, a Bag-Of-Words of PPMI (BOW-PPMI) scores and a standard Latent Dirichlet Allocation (LDA) Topic Model. As well as the original filtering done to the representations, for the BOW-PPMI additionally all terms are filtered out that do not occur in at least  $(d_N/1000)$  documents. Otherwise, there would be too many irrelevant terms to be a fair comparison. The dimension amounts that are compared are of size (50, 100, 200). The MDS space is not available for sentiment, as the memory cost was too prohibitive with 50,000 documents, and there are no doc2vec spaces for placetypes/movies, as it was only possible access to the Bag-Of-Words representation.

When obtaining the single word directions, starting with all of the baseline representations and vocabularies, the infrequent terms are filtered from these vocabularies according to a hyper-parameter that is tuned. As the doc2vec has already been hyper-parameter optimized, the optimal doc2vec space that scored the highest for its class on a Linear SVM is used, rather than tuning the entire process around the doc2vecs vectors. So for example, when evaluating the Keywords task for the movies, directions are obtained from the doc2vec space that performed

best for a linear SVM on the Keywords task following the previous experiments.

Results are obtained for the rankings induced from these word directions on Decision Tree's limited to a depth-three in-order to select the best parameters when using directions for each class. The parameters that are desirable to determine are the type of Vector Space Model, the size of the space, the frequency threshold and the score threshold, which determines the top scoring directions. To do so, for each space-type of each size, a grid search is used to find the best frequency and score cut-offs for that sized space-type. Then, from these space-types and sizes the best performing one is selected. There is a balance between finding words which are useful for creating salient features in our clustering step without including too many words which do not. As our clustering methods are unsupervised, it is important that to try and limit the amount of junk being entered into them, despite the classifiers that use these directions typically being able to filter out those directions which are not suitable to the class. Additionally, as the vocabulary size varies from dataset to dataset, the threshold will naturally be different for each one.

These results allow us to choose for each class, the best Vector Space Model and Scoring-type for that class. Next, we test single directions, attempting to find a good amount of directions to cluster and not including words which may hamper the unsupervised classification, as well as the best space-type for each domain. We found that generally, classifiers performed better with more data, so we use 20000 as our frequency cutoff and 2000 as our score cutoff. Our hyper-parameters for the frequency cut-off were 5000, 10000 and 20000, and our hyper-parameters for the score-cutoff were 1000 and 2000.

We continue with the optimal space and score-type chosen by the single direction experiments, and use the same frequency and score thresholds as before. Two different clustering algorithms are experimented with: Derrac and K-Means. As these algorithms select centroids from the top-scoring directions or randomly, we can expect that some clusters may not be salient features of the space. This is because top-scoring directions, e.g. for accuracy could simply infrequent terms that do not have much meaning, and these infrequent terms could also be randomly selected. We could use grid-search on the frequency and score cutoffs when obtaining these results in order to avoid terms that may disrupt existing clusters or form cluster centers that are not salient features of the space, but we chose a more standardized process that would rely on the parameters of the clustering algorithms and the ability of the classifiers to filter out clusters that

are not informative, so as to not make a time-costly grid search a necessary part of the process.

For K-means clustering, we use Mini batch K-means, implemented by scikit-learn <sup>5</sup>, introduced by [30] and kmeans++ to initialize [1]

#### 4.4.2 Summary of all Results

To begin, the original dimensions of the space are compared to the rankings on single words, the rankings on cluster directions, and the Bag-Of-Words of PPMI scores and topic models on low-depth Decision Trees. Single directions or clusters outperform the baselines in most cases, with the exceptions being in the place-types domain and the keywords task for the movies. For the keywords task, the natural explanation is that in a depth-1 tree, finding words which are directly corresponding to particular keywords is easier with words than if using directions, not only because certain words may have been filtered out, but also because as they are infrequent they may not be well-represented in the space. In this case, the PPMI representation is perfect, as it can find 1-1 matches with the classes without the representations of those words being spatially influenced by other similar words, as it can be expected for them to be in the space. However, this changes when going from depth-one to depth-two and depth-three, which is likely due to overfitting in the case of the PPMI representation. Sometimes Decision Trees of depth-two outperform those of depth-one, but generally depth-three trees perform best. In the case of the place-types, although topic models and PPMI representations are indeed the best, it is not by a wide-margin. Meanwhile when the single directions perform the best in these domains for other tree types they perform much better than the other approaches. Additionally, place-types is our most unbalanced domain with the least documents, so it is possible that they overfit.

---

<sup>5</sup><https://scikit-learn.org/stable/modules/generated/sklearn.cluster.MiniBatchKMeans.html>

| Genres            |              | Keywords     |              |              | Ratings      |              |              |              |              |
|-------------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| Movies            | D1           | D2           | D3           | D1           | D2           | D3           | D1           | D2           | D3           |
| Space             | 0.301        | 0.358        | 0.354        | 0.185        | 0.198        | 0.201        | 0.463        | 0.475        | 0.486        |
| Single directions | <b>0.436</b> | 0.463        | 0.492        | 0.23         | <b>0.233</b> | <b>0.224</b> | 0.466        | 0.499        | 0.498        |
| Clusters          | 0.431        | <b>0.513</b> | <b>0.506</b> | 0.215        | 0.22         | 0.219        | <b>0.504</b> | <b>0.507</b> | <b>0.513</b> |
| PPMI              | 0.429        | 0.443        | 0.483        | <b>0.243</b> | 0.224        | 0.224        | 0.47         | 0.453        | 0.453        |
| Topic             | 0.415        | 0.472        | 0.455        | 0.189        | 0.05         | 0.075        | 0.473        | 0.243        | 0.38         |
| Newsgroups        |              | Sentiment    |              |              | Reuters      |              |              |              |              |
| D1                | D2           | D3           | D1           | D2           | D3           | D1           | D2           | D3           |              |
| Rep               | 0.251        | 0.366        | 0.356        | 0.705        | 0.77         | 0.773        | 0.328        | 0.413        | 0.501        |
| Single dir        | 0.418        | <b>0.49</b>  | <b>0.537</b> | 0.784        | 0.814        | <b>0.821</b> | <b>0.678</b> | <b>0.706</b> | 0.72         |
| Cluster           | 0.394        | 0.433        | 0.513        | 0.735        | <b>0.844</b> | 0.813        | 0.456        | 0.569        | 0.583        |
| PPMI              | 0.33         | 0.407        | 0.444        | 0.7          | 0.719        | 0.73         | 0.616        | 0.699        | <b>0.723</b> |
| Topic             | <b>0.431</b> | 0.423        | 0.444        | <b>0.79</b>  | 0.791        | 0.811        | 0.411        | 0.527        | 0.536        |
| Foursquare        |              | OpenCYC      |              |              | Geonames     |              |              |              |              |
| Placetypes        | D1           | D2           | D3           | D1           | D2           | D3           | D1           | D2           | D3           |
| Rep               | 0.438        | 0.478        | 0.454        | 0.383        | 0.397        | 0.396        | 0.349        | 0.34         | 0.367        |
| Single dir        | <b>0.541</b> | 0.498        | <b>0.531</b> | 0.404        | <b>0.428</b> | 0.39         | <b>0.444</b> | <b>0.533</b> | <b>0.473</b> |
| Cluster           | 0.462        | 0.507        | 0.496        | <b>0.413</b> | 0.42         | <b>0.429</b> | 0.444        | 0.458        | 0.47         |
| PPMI              | 0.473        | <b>0.512</b> | 0.491        | 0.371        | 0.351        | 0.352        | 0.361        | 0.301        | 0.242        |
| Topic             | 0.488        | 0.433        | 0.526        | 0.365        | 0.271        | 0.313        | 0.365        | 0.3          | 0.219        |

Table 4.6: summary of all results

### 4.4.3 Baseline Representations

In Table 4.7 all variations of the baseline representations used directly as input to Decision Trees and SVM's are shown. These examples that do not apply our methodology, serve as a reference point for what is possible using standard linear models without the need for interpretability. In the representations, there is a big performance drop when going from depth three trees to depth one trees. These kind of performance drops are expected for these representations, as they do not have dimensions that correspond to key semantics, so it is unlikely that a smaller tree can use the available dimensions to model a class with limited depth. In this full table the precision and recall scores are included for clarity, mainly to explain why the high recall scores occur. This is because the weights are balanced as a hyper-parameters, and when the weight is balanced so that positive instances are weighted more heavily, the model prioritizes recall over precision. When this high recall score doesn't occur, that means that not balancing the weights performed better on the development data.

The size of the space is not as influential as the representation type in these results for the Decision Trees. For this reason only the best performing representation of each type are shown in Table 4.7. Out of the space-types, PCA performed much better than its counterparts for reuters, newsgroups and sentiment. The MDS representation performs comparably well using a unrestricted depth tree or an SVM, which shows that with a classifier that can make use of all the dimensions, the performance does not decrease as much. This is likely due to the way that PCA orders its dimensions in importance, resulting in key semantics in its first dimensions, giving it an advantage in low-depth Decision Trees. However, this does not necessarily mean that it contains better directions. In the single directions results, PCA is outperformed by MDS and other representations in F1 score for low Decision Tree depths in any of these domains, with the exception of the depth-two trees for sentiment. Despite MDS often encoding the key semantics across more dimensions than other representations, our method is still able find meaningful directions from this space. There is little link between performance on the raw dimensions of the space and performance with rankings on directions in low-depth Decision Trees. This is somewhat counterintuitive, as it would be normal to expect that a representation which performs poorly when used directly as input to a classifier would have similar performance after a linear transformation, but the reason that it works in our case is because low-depth Decision Trees rely on key semantics being disentangled into individual dimensions. Despite the information

---

encoded in the space, if it is not disentangled then the classifier will not perform well.

| Newsgroups | D1           |              |              | D2           |             |              | D3           |              |              | DN           |              |              | SVM          |              |              |
|------------|--------------|--------------|--------------|--------------|-------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
|            | ACC          | F1           | Prec         | Rec          | ACC         | F1           | Prec         | Rec          | ACC          | F1           | Prec         | Rec          | ACC          | F1           | Rec          |
| PCA 200    | 0.701        | 0.251        | 0.148        | 0.811        | 0.843       | 0.366        | 0.245        | 0.719        | 0.956        | 0.355        | 0.54         | 0.265        | 0.946        | 0.44         | 0.45         |
| PCA 100    | 0.698        | 0.247        | 0.146        | 0.813        | 0.835       | 0.362        | 0.241        | 0.731        | 0.957        | 0.356        | 0.576        | 0.257        | 0.948        | 0.451        | 0.465        |
| PCA 50     | 0.68         | 0.24         | 0.141        | 0.829        | 0.834       | 0.355        | 0.234        | 0.735        | 0.957        | 0.329        | 0.472        | 0.253        | 0.947        | 0.45         | 0.462        |
| AWV 200    | 0.687        | 0.217        | 0.126        | 0.781        | 0.758       | 0.256        | 0.156        | 0.718        | 0.764        | 0.26         | 0.157        | 0.751        | 0.937        | 0.339        | 0.352        |
| AWV 100    | 0.677        | 0.21         | 0.122        | 0.775        | 0.78        | 0.275        | 0.173        | 0.683        | 0.746        | 0.25         | 0.149        | 0.769        | 0.934        | 0.324        | 0.332        |
| AWV 50     | 0.696        | 0.219        | 0.127        | 0.772        | 0.777       | 0.272        | 0.168        | 0.71         | 0.743        | 0.25         | 0.149        | 0.786        | 0.935        | 0.325        | 0.335        |
| MDS 200    | 0.581        | 0.184        | 0.103        | <b>0.837</b> | 0.742       | 0.262        | 0.16         | 0.729        | 0.719        | 0.236        | 0.139        | 0.785        | 0.935        | 0.327        | 0.332        |
| MDS 100    | 0.586        | 0.187        | 0.105        | 0.833        | 0.754       | 0.261        | 0.159        | 0.727        | 0.705        | 0.236        | 0.138        | <b>0.808</b> | 0.935        | 0.33         | 0.338        |
| MDS 50     | 0.593        | 0.153        | 0.087        | 0.647        | 0.716       | 0.25         | 0.15         | <b>0.756</b> | 0.736        | 0.243        | 0.144        | 0.774        | 0.935        | 0.324        | 0.335        |
| D2V 200    | 0.682        | 0.205        | 0.119        | 0.746        | 0.802       | 0.268        | 0.169        | 0.646        | 0.77         | 0.269        | 0.164        | 0.75         | 0.94         | 0.366        | 0.389        |
| D2V 100    | 0.682        | 0.208        | 0.12         | 0.762        | 0.792       | 0.268        | 0.168        | 0.662        | 0.786        | 0.268        | 0.164        | 0.727        | 0.94         | 0.376        | 0.392        |
| D2V 50     | 0.683        | 0.207        | 0.12         | 0.764        | 0.809       | 0.294        | 0.187        | 0.694        | 0.782        | 0.28         | 0.172        | 0.761        | 0.943        | 0.394        | 0.415        |
| PPMI       | <b>0.948</b> | 0.33         | <b>0.532</b> | 0.239        | 0.947       | 0.407        | 0.511        | 0.338        | 0.944        | <b>0.444</b> | 0.506        | 0.396        | <b>0.951</b> | <b>0.494</b> | <b>0.496</b> |
| Topic      | 0.852        | <b>0.431</b> | 0.304        | 0.743        | <b>0.96</b> | <b>0.423</b> | <b>0.604</b> | 0.326        | <b>0.961</b> | 0.444        | <b>0.606</b> | 0.35         | 0.944        | 0.432        | 0.434        |
|            |              |              |              |              |             |              |              |              |              |              |              |              | 0.879        | 0.46         | 0.318        |
|            |              |              |              |              |             |              |              |              |              |              |              |              | 0.962        | 0.613        | 0.627        |
|            |              |              |              |              |             |              |              |              |              |              |              |              | <b>0.492</b> | <b>0.496</b> | <b>0.835</b> |

Table 4.7: Full results for the newsgroups.

Table 4.8: Results for all other domains for the representations.

| Reuters    | D1           |              | D2           |              | D3           |              | Sentiment    |              | D1           |              | D2           |              | D3           |              | DN           |              | SVM          |              |
|------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
|            | ACC          | F1           | ACC          | F1           | ACC          | F1           | ACC          | F1           | ACC          | F1           | ACC          | F1           | ACC          | F1           | ACC          | F1           | ACC          | F1           |
| PCA        | 0.847        | 0.328        | 0.917        | 0.413        | 0.978        | 0.501        | 0.978        | 0.565        | 0.989        | 0.761        | 0.745        | 0.705        | 0.777        | 0.778        | 0.773        | <b>0.781</b> | <b>0.779</b> | <b>0.893</b> |
| AWV        | 0.782        | 0.252        | 0.971        | 0.328        | 0.974        | 0.417        | 0.973        | 0.495        | 0.987        | 0.719        | 0.642        | 0.652        | 0.643        | 0.694        | 0.695        | 0.717        | 0.66         | 0.829        |
| MDS        | 0.791        | 0.263        | 0.9          | 0.357        | 0.979        | 0.489        | 0.976        | 0.522        | 0.988        | 0.67         | 0.642        | 0.664        | 0.66         | 0.707        | 0.702        | 0.7          | 0.711        | 0.878        |
| D2V        | 0.818        | 0.268        | 0.867        | 0.298        | 0.974        | 0.445        | 0.971        | 0.482        | 0.986        | 0.724        | 0.616        | 0.7          | 0.655        | 0.719        | 0.675        | 0.73         | 0.712        | 0.888        |
| PPMI       | <b>0.975</b> | <b>0.616</b> | <b>0.978</b> | <b>0.699</b> | <b>0.98</b>  | <b>0.723</b> | <b>0.984</b> | <b>0.746</b> | <b>0.99</b>  | <b>0.8</b>   | <b>0.793</b> | <b>0.79</b>  | <b>0.794</b> | <b>0.791</b> | <b>0.81</b>  | <b>0.811</b> | 0.73         | 0.822        |
| Topic      | 0.92         | 0.411        | 0.977        | 0.527        | 0.977        | 0.536        | 0.977        | 0.56         | 0.95         | 0.513        |              |              |              |              |              |              |              |              |
| Placetypes | D1           |              | D2           |              | D3           |              | DN           |              | SVM          |              | D1           |              | D2           |              | D3           |              | DN           |              |
| OpenCYC    | ACC          | F1           | ACC          | F1           | ACC          | F1           | ACC          | F1           | ACC          | F1           | ACC          | F1           | ACC          | F1           | ACC          | F1           | ACC          | F1           |
| PCA        | 0.586        | 0.346        | 0.708        | 0.343        | 0.695        | 0.342        | 0.832        | 0.309        | 0.847        | 0.474        | 0.722        | 0.301        | 0.755        | 0.339        | 0.717        | 0.321        | 0.884        | 0.518        |
| AWV        | 0.625        | <b>0.383</b> | 0.651        | 0.376        | 0.728        | <b>0.396</b> | <b>0.844</b> | <b>0.362</b> | 0.85         | 0.466        | 0.679        | 0.29         | 0.774        | 0.321        | 0.756        | 0.343        | 0.873        | 0.496        |
| MDS        | 0.624        | 0.364        | 0.7          | <b>0.397</b> | 0.731        | 0.374        | 0.843        | 0.305        | 0.861        | <b>0.476</b> | 0.679        | 0.298        | 0.79         | 0.358        | 0.773        | 0.354        | 0.887        | <b>0.532</b> |
| PPMI       | <b>0.728</b> | 0.371        | 0.75         | 0.351        | 0.739        | 0.352        | 0.843        | 0.323        | <b>0.9</b>   | 0.366        | <b>0.852</b> | <b>0.429</b> | <b>0.91</b>  | 0.443        | <b>0.912</b> | <b>0.483</b> | <b>0.416</b> | 0.526        |
| Topic      | 0.708        | 0.365        | <b>0.87</b>  | 0.271        | <b>0.87</b>  | 0.313        | 0.831        | 0.313        | 0.808        | 0.407        | 0.767        | 0.415        | 0.905        | <b>0.472</b> | 0.912        | 0.455        | <b>0.889</b> | 0.491        |
| Placetypes | D1           |              | D2           |              | D3           |              | DN           |              | SVM          |              | D1           |              | D2           |              | D3           |              | DN           |              |
| Foursquare | ACC          | F1           | ACC          | F1           | ACC          | F1           | ACC          | F1           | ACC          | F1           | ACC          | F1           | ACC          | F1           | ACC          | F1           | ACC          | F1           |
| PCA        | 0.731        | 0.342        | 0.823        | 0.393        | 0.86         | 0.388        | 0.887        | 0.398        | 0.896        | 0.568        | 0.647        | 0.185        | 0.644        | 0.193        | 0.677        | 0.199        | 0.846        | 0.272        |
| AWV        | 0.767        | 0.401        | 0.828        | 0.478        | 0.85         | 0.452        | 0.905        | <b>0.505</b> | 0.923        | <b>0.622</b> | 0.5          | 0.16         | 0.641        | 0.179        | 0.595        | 0.174        | 0.853        | 0.23         |
| MDS        | <b>0.915</b> | 0.438        | 0.804        | 0.427        | 0.86         | 0.454        | 0.893        | 0.462        | 0.932        | 0.619        | 0.633        | 0.179        | 0.69         | 0.198        | 0.674        | 0.201        | 0.84         | <b>0.28</b>  |
| PPMI       | 0.889        | 0.473        | 0.915        | <b>0.512</b> | 0.904        | 0.491        | 0.881        | 0.31         | <b>0.938</b> | 0.567        | <b>0.818</b> | <b>0.243</b> | 0.745        | <b>0.224</b> | 0.739        | <b>0.224</b> | 0.847        | 0.217        |
| Topic      | 0.864        | <b>0.488</b> | <b>0.916</b> | 0.433        | <b>0.917</b> | <b>0.526</b> | <b>0.907</b> | 0.464        | 0.916        | 0.569        | 0.629        | 0.189        | <b>0.932</b> | 0.05         | <b>0.93</b>  | 0.075        | <b>0.857</b> | 0.21         |
| Placetypes | D1           |              | D2           |              | D3           |              | DN           |              | SVM          |              | D1           |              | D2           |              | D3           |              | DN           |              |
| Geonames   | ACC          | F1           | ACC          | F1           | ACC          | F1           | ACC          | F1           | ACC          | F1           | ACC          | F1           | ACC          | F1           | ACC          | F1           | ACC          | F1           |
| PCA        | 0.502        | 0.301        | 0.69         | 0.305        | 0.68         | 0.295        | 0.821        | 0.243        | 0.844        | 0.401        | <b>0.65</b>  | 0.463        | 0.681        | <b>0.475</b> | 0.684        | <b>0.486</b> | 0.744        | 0.58         |
| AWV        | 0.657        | 0.326        | 0.755        | 0.323        | 0.842        | <b>0.367</b> | 0.813        | 0.332        | 0.865        | <b>0.514</b> | 0.601        | 0.423        | 0.618        | 0.433        | 0.596        | 0.448        | 0.736        | 0.532        |
| MDS        | 0.626        | 0.349        | 0.695        | <b>0.34</b>  | 0.796        | 0.272        | <b>0.845</b> | 0.295        | 0.638        | 0.397        | 0.592        | 0.437        | 0.635        | 0.449        | 0.631        | 0.452        | <b>0.752</b> | <b>0.589</b> |
| PPMI       | <b>0.808</b> | 0.361        | 0.732        | 0.301        | 0.76         | 0.242        | 0.83         | 0.283        | <b>0.894</b> | 0.312        | 0.583        | 0.47         | 0.635        | 0.453        | 0.605        | 0.453        | 0.73         | 0.536        |
| Topic      | 0.771        | <b>0.365</b> | <b>0.863</b> | 0.3          | <b>0.85</b>  | 0.219        | 0.828        | <b>0.348</b> | 0.819        | 0.349        | 0.575        | <b>0.473</b> | <b>0.789</b> | 0.243        | <b>0.789</b> | 0.38         | 0.739        | 0.501        |



#### 4.4.4 Word Directions

Although Linear SVM's perform the best on these representations without the need for interpretability, other results will be for low-depth Decision Trees in-order to easily distinguish the degree to which key semantics correspond to dimensions in the representations.

The main takeaway from this section is that in most cases performance greatly increases compared to the original representations used directly as input to the model (For the exact differences, see Appendix 7.1).

Interestingly, there was also more variance in the difference between space-type sizes, making it an important hyper-parameter for the single directions. The best space type also varied across domains. Loosely, it is possible to attribute the performance increase for a space-type to either modelling the rankings for the same directions better, or containing unique terms that were particularly relevant to the classes. However, when looking at the qualitative results, generally the words common to all space-types are the most salient 4.3. We can see if this is the case by looking at the Decision Trees for the same task that had the most difference between the space-types and space-sizes. If a Decision Tree contains mostly similar words, but the performance is greater, we can attribute it to a better quality ranking in the space. If the Decision Tree contains different words, especially as the first node, then we know that it was because the words that were modelled well were different between them.

We see that generally, the best space type is the same across a variety of tasks in the same domain, AWV is the best for the place-types but MDS is best for the movies (despite a marginal difference in the ratings). This could mean that performance on one natural task will generalize well to the others, so the space-type/size of the space that we identify contains the key semantics for that domain rather than a particular task.

NDCG was selected as the best score-type for Sentiment, Newsgroups, Reuters, Movies Genres, Movies Keywords in depth-3 Decision Trees. Place-types foursquare used F1-score, but the classes are very unbalanced and there are few documents.

| Newsgroups | D1           |              |              |              | D2           |              |              |              | D3           |              |              |              |
|------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
|            | ACC          | F1           | Prec         | Rec          | ACC          | F1           | Prec         | Rec          | ACC          | F1           | Prec         | Rec          |
| PCA 200    | 0.955        | 0.348        | 0.521        | 0.261        | 0.959        | 0.424        | 0.678        | 0.309        | 0.96         | 0.454        | 0.674        | 0.343        |
| PCA 100    | 0.957        | 0.382        | 0.491        | 0.313        | 0.961        | 0.474        | 0.679        | 0.364        | <b>0.963</b> | 0.512        | 0.694        | 0.406        |
| PCA 50     | 0.957        | 0.373        | 0.417        | 0.337        | <b>0.963</b> | 0.478        | 0.621        | 0.388        | 0.963        | 0.506        | 0.7          | 0.396        |
| AWV 200    | 0.832        | 0.35         | 0.226        | 0.777        | 0.957        | 0.383        | 0.517        | 0.305        | 0.958        | 0.445        | 0.598        | 0.354        |
| AWV 100    | 0.83         | 0.343        | 0.219        | 0.785        | 0.823        | 0.36         | 0.233        | <b>0.792</b> | 0.956        | 0.387        | 0.563        | 0.295        |
| AWV 50     | 0.807        | 0.341        | 0.215        | 0.816        | 0.833        | 0.361        | 0.236        | 0.762        | 0.954        | 0.392        | 0.511        | 0.318        |
| MDS 200    | <b>0.959</b> | <b>0.418</b> | <b>0.543</b> | 0.339        | 0.962        | 0.465        | 0.669        | 0.357        | 0.962        | 0.493        | <b>0.707</b> | 0.379        |
| MDS 100    | 0.857        | 0.365        | 0.244        | 0.725        | 0.959        | 0.428        | 0.624        | 0.326        | 0.96         | 0.453        | 0.644        | 0.349        |
| MDS 50     | 0.821        | 0.324        | 0.206        | 0.762        | 0.842        | 0.386        | 0.258        | 0.77         | 0.957        | 0.398        | 0.596        | 0.299        |
| D2V 200    | 0.831        | 0.343        | 0.22         | 0.784        | 0.96         | 0.47         | <b>0.683</b> | 0.358        | 0.962        | 0.494        | 0.69         | 0.385        |
| D2V 100    | 0.844        | 0.374        | 0.243        | 0.803        | 0.961        | <b>0.49</b>  | 0.642        | 0.396        | 0.962        | 0.517        | 0.67         | 0.421        |
| D2V 50     | 0.845        | 0.388        | 0.252        | <b>0.844</b> | 0.962        | 0.488        | 0.639        | 0.395        | 0.963        | <b>0.537</b> | 0.673        | <b>0.446</b> |
| Sentiment  |              |              |              |              |              |              |              |              |              |              |              |              |
| Reuters    | D1           | D2           | D3           | D1           | D2           | D3           | D1           | D2           | D3           | D1           | D2           | D3           |
|            | ACC          | F1           | ACC          | F1           | ACC          | F1           | ACC          | F1           | ACC          | F1           | ACC          | F1           |
| PCA        | 0.976        | 0.658        | 0.979        | 0.679        | 0.977        | 0.467        | PCA          | 0.739        | 0.759        | <b>0.797</b> | <b>0.814</b> | 0.802        |
| AWV        | 0.975        | 0.598        | 0.979        | 0.656        | 0.98         | 0.66         | AWV          | 0.7          | 0.699        | 0.711        | 0.736        | 0.735        |
| MDS        | 0.975        | <b>0.678</b> | <b>0.98</b>  | <b>0.706</b> | <b>0.982</b> | <b>0.72</b>  | D2V          | <b>0.776</b> | <b>0.784</b> | 0.782        | 0.801        | <b>0.822</b> |
| D2V        | <b>0.977</b> | 0.583        | 0.979        | 0.664        | 0.98         | 0.632        |              |              |              |              |              | <b>0.821</b> |
| Movies     |              |              |              |              |              |              |              |              |              |              |              |              |
| Placetypes | D1           | D2           | D3           | D1           | D2           | D3           | D1           | D2           | D3           | D1           | D2           | D3           |
|            | ACC          | F1           | ACC          | F1           | ACC          | F1           | ACC          | F1           | ACC          | F1           | ACC          | F1           |
| OpenCYC    | 0.632        | 0.371        | 0.704        | 0.381        | 0.735        | 0.365        | Genres       | 0.824        | 0.412        | 0.82         | 0.441        | 0.913        |
| PCA        | <b>0.66</b>  | <b>0.404</b> | <b>0.734</b> | <b>0.428</b> | <b>0.755</b> | <b>0.39</b>  | PCA          | 0.81         | 0.421        | 0.837        | 0.436        | 0.912        |
| AWV        | 0.658        | 0.374        | 0.711        | 0.385        | 0.746        | 0.35         | MDS          | <b>0.849</b> | <b>0.446</b> | <b>0.839</b> | <b>0.463</b> | <b>0.918</b> |
| MDS        | 0.658        | 0.374        | 0.711        | 0.385        | 0.746        | 0.35         |              |              |              |              |              | <b>0.495</b> |
| Keywords   |              |              |              |              |              |              |              |              |              |              |              |              |
| Foursquare | ACC          | F1           | ACC          | F1           | ACC          | F1           | Keywords     | ACC          | F1           | ACC          | F1           | ACC          |
| PCA        | 0.785        | 0.477        | <b>0.907</b> | 0.474        | 0.869        | <b>0.531</b> | PCA          | 0.737        | 0.225        | 0.727        | 0.227        | <b>0.709</b> |
| AWV        | <b>0.918</b> | <b>0.541</b> | 0.881        | <b>0.498</b> | 0.889        | 0.466        | AWV          | 0.656        | 0.201        | 0.672        | 0.203        | 0.652        |
| MDS        | 0.82         | 0.416        | 0.879        | 0.482        | <b>0.897</b> | 0.485        | MDS          | <b>0.745</b> | <b>0.23</b>  | <b>0.74</b>  | <b>0.233</b> | 0.708        |
| Ratings    |              |              |              |              |              |              |              |              |              |              |              |              |
| Geonames   | ACC          | F1           | ACC          | F1           | ACC          | F1           | Ratings      | ACC          | F1           | ACC          | F1           | ACC          |
| PCA        | 0.665        | 0.348        | 0.754        | 0.342        | 0.743        | 0.306        | PCA          | <b>0.647</b> | <b>0.466</b> | <b>0.721</b> | <b>0.499</b> | 0.681        |
| AWV        | <b>0.711</b> | <b>0.444</b> | <b>0.795</b> | <b>0.533</b> | <b>0.802</b> | <b>0.473</b> | AWV          | 0.646        | 0.463        | 0.692        | 0.474        | 0.677        |
| MDS        | 0.591        | 0.289        | 0.772        | 0.333        | 0.764        | 0.352        | MDS          | 0.62         | 0.463        | 0.692        | 0.489        | <b>0.686</b> |

Table 4.9: all dirs

### 4.4.5 Clustered Directions

?? These results were obtained by taking the single directions that performed the best in the previous results and clustering them with a variety of hyper-parameters for the clusters. K-means mostly outperforms Derrac. It does not in the case of Keywords, where it performs better for every Decision Tree. Although the differences in absolute values are quite small in this case, it is still significant as it is quite difficult to achieve high performance on this task, making these relative changes important. This case can give us insight into how disentanglement affects performance on different classes and domains - and how our unsupervised method selects the best parameters.

When looking into the how the individual classes fared, the 100-size Derrac clusters performed better at the keywords "shot-in-the-chest" and "machine-gun" and sacrificed performance in the "sequel" class. In Derrac, there was the following cluster ("soldiers combat fighting military battle ... weapons rambo gunfights spaghetti guns ...") while in the best performing k-means 200-size clusters these words were split into two separate clusters, one for guns ("gun explosions shoot shooting weapons ... rambo") and one for military ("war soldiers combat military ... platoon infantry"). It's possible that as the Derrac method combined these together into their own cluster they were able to better capture the classes for "shot-in-the-chest" and "machine-guns" because these things occurred in war films where people were shot or shooting. So in this case, the parameters chosen for Derrac supported the classification of the documents into keywords because they better captured particular class concepts through a lesser degree of disentanglement. This idea is supported when looking at the depth-three tree for this class, which uses this cluster as its first node as well as a node in the depth-two layer. This is an instance where having a heavily populated cluster average their direction performs better than strongly disentangling the concepts.

Meanwhile, this same lack of disentanglement caused it to lose performance in the "sequel" class. In K-means, the cluster was found for ("franchise sequels sequel installments") while in Derrac the cluster was ("franchise sequels sequel instalments entry returns"). This cluster was also chosen in Derrac as the first node of its Decision Tree, but this caused it to perform worse than k-means. This is likely because although the words "entry" and "returns" were most similar to this cluster, they disrupted the direction too much. Indeed, when looking at the k-

means clusters, the "returns" direction is clustered with "events situation conclusion spoiler ... protagonists exscapes break scenario ...", seemingly referring to a character or thing "returning" in a conclusive part of the movie, and the word "entry" is clustered with the words "effective genuine ... hits build surprisingly ... succeeds essentially finale entry ..." seemingly relating to a more sentiment related cluster about how a movie performed. So in this case k-means being able to find more disentangled clusters than Derrac gave it a performance advantage.

This could be due to the best-performing Derrac clusters being 100-size (meaning the clusters would contain more terms) and the k-means being 200-size. However, in the 100-size K-means clusters, "gun" and "explosions" ended up being in a cluster with ("western outlaw heist shootout west"), making it a more western oriented cluster, and the idea of a war was even more disentangled with a single cluster corresponding to ("war soldiers military soldier army sergeant sgt platoon infantry"). In conclusion, Derrac for the Keywords task captured certain concepts better than k-means, in particular by clustering together the idea of "war" and "guns" to achieve high performance on the keywords "shot-in-the-chest" and "machine-guns". K-means favoured a more disentangled approach to these ideas, which meant that although it captured the idea of "war" well, it was not able to capture the classes inbetween the idea of "war" and "guns".

In conclusion, the clustering method that performs the best for a task in this unsupervised context is the one that creates clusters that correspond closely with the task's classes, through clustering together words which average into a particular concept, or disentangling words into concepts so that they more precisely model it.

| Newsgroups  | D1           |              |              | D2           |              |              | D3          |             |              |              |              |              |
|-------------|--------------|--------------|--------------|--------------|--------------|--------------|-------------|-------------|--------------|--------------|--------------|--------------|
|             | ACC          | F1           | Prec         | Rec          | ACC          | F1           | Prec        | Rec         | ACC          | F1           | Prec         | Rec          |
| K-means 200 | <b>0.852</b> | <b>0.394</b> | <b>0.261</b> | 0.795        | <b>0.958</b> | <b>0.433</b> | <b>0.58</b> | 0.345       | <b>0.963</b> | <b>0.513</b> | <b>0.704</b> | 0.403        |
| K-means 100 | 0.842        | 0.388        | 0.257        | 0.791        | 0.958        | 0.366        | 0.516       | 0.284       | 0.962        | 0.5          | 0.635        | <b>0.412</b> |
| K-means 50  | 0.834        | 0.381        | 0.248        | <b>0.819</b> | 0.815        | 0.336        | 0.212       | <b>0.81</b> | 0.961        | 0.485        | 0.612        | 0.402        |
| Derrac 200  | 0.803        | 0.313        | 0.202        | 0.693        | 0.797        | 0.306        | 0.191       | 0.781       | 0.958        | 0.409        | 0.605        | 0.309        |
| Derrac 100  | 0.792        | 0.305        | 0.197        | 0.667        | 0.791        | 0.287        | 0.179       | 0.721       | 0.957        | 0.374        | 0.56         | 0.281        |
| Derrac 50   | 0.769        | 0.26         | 0.162        | 0.661        | 0.768        | 0.237        | 0.143       | 0.693       | 0.955        | 0.315        | 0.47         | 0.237        |

Table 4.10: All clustering size results for the newsgroups

| Reuters    | D1           |              | D2           |              | D3           |              | Sentiment |              | D1           |              | D2           |              | D3           |    |
|------------|--------------|--------------|--------------|--------------|--------------|--------------|-----------|--------------|--------------|--------------|--------------|--------------|--------------|----|
|            | ACC          | F1           | ACC          | F1           | ACC          | F1           | ACC       | F1           | ACC          | F1           | ACC          | F1           | ACC          | F1 |
| K-means    | <b>0.875</b> | <b>0.338</b> | <b>0.975</b> | <b>0.54</b>  | 0.973        | <b>0.58</b>  | K-means   | 0.623        | 0.674        | <b>0.837</b> | <b>0.844</b> | 0.658        | 0.707        |    |
| Derrac     | 0.797        | 0.291        | 0.973        | 0.402        | <b>0.974</b> | 0.485        | Derrac    | <b>0.712</b> | <b>0.735</b> | 0.802        | 0.82         | <b>0.803</b> | <b>0.813</b> |    |
| Placetypes | D1           |              | D2           |              | D3           |              | Movies    | D1           |              | D2           |              | D3           |              |    |
| OpenCYC    | ACC          | F1           | ACC          | F1           | ACC          | F1           | Genres    | ACC          | F1           | ACC          | F1           | ACC          | F1           |    |
| K-means    | <b>0.641</b> | <b>0.413</b> | <b>0.735</b> | <b>0.405</b> | 0.75         | <b>0.43</b>  | K-means   | <b>0.813</b> | <b>0.431</b> | <b>0.913</b> | <b>0.513</b> | <b>0.913</b> | <b>0.506</b> |    |
| Derrac     | 0.605        | 0.39         | 0.672        | 0.392        | <b>0.755</b> | 0.391        | Derrac    | 0.759        | 0.341        | 0.789        | 0.431        | 0.911        | 0.432        |    |
| Foursquare | ACC          | F1           | ACC          | F1           | ACC          | F1           | Keywords  | ACC          | F1           | ACC          | F1           | ACC          | F1           |    |
| K-means    | <b>0.913</b> | <b>0.462</b> | <b>0.911</b> | <b>0.5</b>   | <b>0.891</b> | <b>0.511</b> | K-means   | 0.667        | 0.208        | 0.648        | 0.202        | 0.678        | 0.213        |    |
| Derrac     | 0.768        | 0.392        | 0.835        | 0.445        | 0.805        | 0.425        | Derrac    | <b>0.726</b> | <b>0.215</b> | <b>0.745</b> | <b>0.22</b>  | <b>0.707</b> | <b>0.219</b> |    |
| Geonames   | ACC          | F1           | ACC          | F1           | ACC          | F1           | Ratings   | ACC          | F1           | ACC          | F1           | ACC          | F1           |    |
| K-means    | <b>0.772</b> | 0.43         | <b>0.774</b> | 0.407        | <b>0.819</b> | <b>0.472</b> | K-means   | <b>0.671</b> | <b>0.504</b> | 0.638        | <b>0.507</b> | <b>0.686</b> | <b>0.513</b> |    |
| Derrac     | 0.678        | <b>0.449</b> | 0.74         | <b>0.411</b> | 0.807        | 0.415        | Derrac    | 0.651        | 0.445        | <b>0.669</b> | 0.463        | 0.627        | 0.479        |    |

Table 4.11: The best clustering results for each domain and task

### 4.4.6 Conclusion

In conclusion, we introduce a methodology to go from a Vector Space Model of Semantics and an associated bag-of-words to an interpretable representation and interpretable classifiers. We define an interpretable representation in this work as having two properties: disentanglement and labels, and an interpretable classifier as a simple linear classifier that has components corresponding to the interpretable representation that has these properties, e.g. nodes in a decision tree. In general, we give a simple methodology that can be used to achieve interpretable features and classifiers as an alternative to methods like Topic Models, and give insight into the parameters required and qualitative results that can be obtained. We extensively test the qualitative and quantitative results, finding that the highest-performing quantitative results also make good intuitive qualitative sense. We find that our method greatly outperforms the original representations on low-depth Decision Trees, giving good evidence that we have disentangled the representation. Additionally, we find that we are also competitive with standard interpretable representation baselines in most cases. We introduce variations to the original work that produced these kind of interpretable representations, in particular finding that scoring directions using NDCG performed better than Kappa in most cases, and that we could achieve much stronger results than the original clustering method using K-means. Further, we experimented using a variety of space-types and domains, verifying that the methodology can be applied more generally than shown in [6]. The main experiments that would be interesting to expand on for this chapter would be more state-of-the-art representations, specific investigations of how those representations are able to achieve such strong results, and interpretability experiments to see how our cluster labels fare in real-world situations.

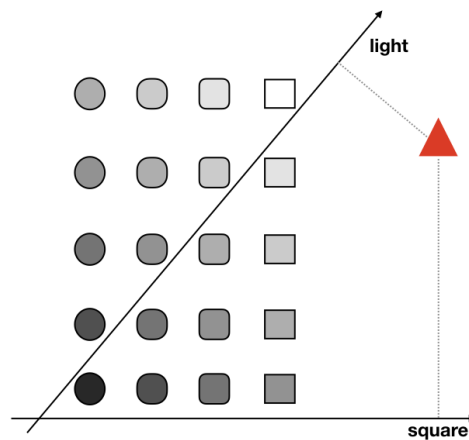
# **Fine-tuning Vector Spaces to Improve Their Directions**

## **5.1 Introduction**

Chapter 4 introduced a method to obtain feature-directions from off-the-shelf vector-spaces, as well as methods to test the quality of these feature-directions and their associated feature-rankings. Then, this method was applied in Chapter 6.1 to obtain feature-directions from the layers of neural networks. However, feature-directions obtained from either of these vector spaces can sometimes be sub-optimal. For example in the case of neural network auto-encoders, it was found that the quality of feature-directions in a auto-encoder representation degrade from a maximal Kappa score of 0.52 in the initial layer of to a maximal Kappa score of 0.18 on the 5th layer (See Section ??).

In figure 5.1, a problem that can occur with feature-directions from representations learned with a similarity-centred objective, e.g. Multi-Dimensional Scaling (see Section ??) is illustrated. This is an example problem in the toy domain of shapes, where basic geometric shapes are embedded in a two-dimensional space. In this example, directions have been identified which encode how light an object is and how closely its shape resembles a square. While most of the shapes embedded in this space are grey-scale circles and squares, one of the shapes embedded in this space is a red triangle, a clear outlier. When considering that the objective the space is learned with is similarity, the spatial representation for this triangle is correct, as it is far from all the other shapes. However, when ranking the shapes on the feature-directions for square and light, the outlier takes up an extreme position on the rankings. This means that the triangle is ranked incorrectly, as it is considered to be the shape that most exhibits the features “light” and





**Figure 5.1:** Toy example showing the effect of outliers in a two-dimensional embedding of geometric shapes..

“square”.

Ideally, representations would be learned with knowledge of feature-directions. For example, the method to learn a representation of the toy domain would know that it should model the features "square" and "light" rather than a similarity objective, so that this triangle would end up closer to the bottom-left corner. However, as we cannot a-priori determine the features the space must be learned from, it is difficult to learn a representation in this way. This Chapter instead introduces an unsupervised method that given a representation and its associated feature-directions, we can obtain a vector space and associated feature-directions where the quality of the feature-directions is prioritized over the existing structure. The intention of this method is to resolve issues like those described in the previous two paragraphs.

To introduce the idea behind the method, we start with the assumption that each feature-direction has one feature-word, which describes the feature: if the feature-ranking of a document on a feature-direction is faithful to the bag-of-words score for the feature-word in the document, then the feature-ranking is good. To give an example of why this assumption is useful, in the IMDB movies domain 3.2 Multi-Dimensional Scaling (MDS) spaces there is the case of an Indian Bollywood movie that is very unlike other movies, as its reviews only use language specific to Bollywood films and the amount of reviews it has is low overall. This movie occupies a top-ranking position in a variety of feature-directions, as a consequence of it being very dissimilar to other movies. The fine-tuning process solves this problem by attempting to match its ranking in the vector space that is very high, to its bag-of-words value, which is zero. This results in this

| Feature direction                       | Highest ranking objects            | Highest fine-tuned ranking objects             |
|---|------------------------------------|--|
| {steep, climb, slope}                   | mountain, landscape, national park | ski slope, steep slope, slope                  |
| {illuminated, illumination, skyscraper} | building, city, skyscraper         | tall building, office building, large building |
| {play, kid, kids}                       | school, field, fence               | college classroom, classroom, school           |
| {spooky, creepy, scary}                 | hallway, fence, building           | hospital room, hospital ward, patient room     |
| {amazing, dream, awesome}               | fence, building, beach             | hotel pool, resort, beach resort               |
| {pavement, streetlight, streets}        | sidewalk, fence, building          | overpass road, overpass, road junction         |
| {dead, hole, death}                     | fence, steps, park                 | grave, cemetery, graveyard                     |
| {spire, belltower, towers}              | building, arch, house              | bell tower, arch, religious site               |
| {stones, moss, worldheritage}           | landscape, fence, steps            | ancient site, ancient wall, tomb               |
| {mosaic, tile, bronze}                  | building, city, steps              | cathedral, church, religious site              |

**Table 5.1: Comparing the highest ranking place-type objects in the original and fine-tuned space. .**

obscure outlier movie being moved down drastically in the rankings.

To give some real examples, in Table 5.1, names of documents are shown ranked on feature directions in the domain of place-types (See Section 3.2). In these examples, for the cluster-feature *{steep, climb, slope}*, the top ranked document *mountain* is clearly relevant. However, the next two documents — *landscape* and *national park* — are not directly related to this feature. Intuitively, they are ranked highly because of their similarity to *mountain* in the vector space. Similarly, for the second feature, *building* is ranked highly because of its similarity to *skyscraper*, despite intuitively not having this feature. Finally, *fence* received a high rank for several features, mostly because it is an outlier in the space.

Generally, the method that fine-tunes vector spaces and their associated feature-directions is as follows: First, a vector space is learned from bag-of-words representations of the considered documents, using a standard similarity-centric method or neural network. Next, the method from Chapter ?? is used to obtain feature-directions and their associated words from a vector space. Then, following our assumption outlined in the previous paragraph, documents are ranked on the feature-direction’s associated words using the bag-of-words. Finally, this ranking is used to fine-tune the vector space and feature-directions so that the resulting feature-rankings are more faithful to the ranking on the bag-of-words.

This Chapter is a follow-up of the previous two Chapters, where previously feature-directions are identified in a variety of vector-spaces, and their potential applications are discussed, this Chapter focuses on improving the quality of these feature-directions to achieve better results. This Chapter continues with explaining the method to fine-tune a vector space and its associated feature directions using a bag-of-words in detail. Afterwards, we show quantitative results to see how the fine-tuning affects simple interpretable classifiers (as in Chapter ??). Finally, we end with a conclusion for potential future work.

## 5.2 Fine-Tuning Vector Spaces And Their Associated Feature Directions

To improve the directions and address these problems, we propose a method for fine-tuning the semantic space representations and corresponding feature directions. First, it is explained how to obtain target rankings from PPMI scores. Then, the neural network that uses these target rankings to improve the vector space and its associated feature-directions is described. The main idea is to use the BoW representations of the objects as a kind of weak supervision signal: if an object should be ranked highly for a given feature, we would expect the words describing that feature to appear frequently in its description. To obtain the target rankings, for each feature  $f$  we determine a total ordering  $\preceq_f$  such that  $o \preceq_f o'$  iff the feature  $f$  is more prominent in the BoW representation of object  $o'$  than in the BoW representation of  $o$ . We will refer to  $\preceq_f$  as the *target ranking* for feature  $f$ . If the feature directions are in perfect agreement with this target ranking, it would be the case that  $o \preceq o'$  iff  $v_C \cdot o \leq v_C \cdot o'$ . Since this will typically not be the case, we subsequently determine *target values* for the dot products  $v_C \cdot o$ . These target values represent the minimal way in which the dot products need to be changed to ensure that they respect the target ranking. Once these rankings have been obtained, we use a simple feedforward neural network to adapt the semantic space representations  $o$  and feature directions  $v_C$  to make the dot products  $v_C \cdot o$  as close as possible to these target values.

### 5.2.1 Generating Target Rankings

Let  $C_1, \dots, C_K$  be the clusters that were found using the method from Section 4.2.1. Each cluster  $C_i$  typically corresponds to a set of semantically related words  $\{w_1, \dots, w_n\}$ , which describe some salient feature from the considered domain. From the BoW representations of the objects, we can now define a ranking that reflects how strongly each object is related to the words from this cluster. To this end, we represent each object as a bag of clusters (BoC) and then compute PPMI scores over this representation. In particular, for a cluster  $C = \{w_1, \dots, w_m\}$ , we define  $n(C, o) = \sum_{i=1}^m n(w_i, o)$ . In other words,  $n(C, o)$  is the total number of occurrences of words from cluster  $C$  in BoW representation of  $o$ . We then write  $ppmi(C, o)$  for the PPMI score corresponding to this BoC representation, which is evaluated in the same way as  $ppmi(C, o)$ , but using the counts  $n(C, o)$  rather than  $n(w, o)$ . The target ranking for cluster  $C_i$  is then such that  $o_1$  is ranked higher than  $o_2$  iff  $ppmi(C_i, o_1) > ppmi(C_i, o_2)$ . By computing PPMI scores w.r.t. clusters of words, we alleviate problems with sparsity and synonymy, which in turn allows us to better estimate the intensity with which a given feature applies to the object. For instance, an object describing a violent movie might not actually mention the word ‘violent’, but would likely mention at least some of the words from the same cluster (e.g. ‘bloody’ ‘brutal’ ‘violence’ ‘gory’). Similarly, this approach allows us to avoid problems with ambiguous word usage; e.g. if a movie is said to contain ‘violent language’, it will not be identified as violent if other words related to this feature are rarely mentioned.

### 5.2.2 Generating Target Feature Values

Finding directions in a vector space that induce a set of given target rankings is computationally hard<sup>1</sup>. Therefore, rather than directly using the target rankings from Section 5.2.1 to fine-tune the semantic space, we will generate target values for the dot products  $v_{C_j} \cdot o_i$  from these target rankings. One straightforward approach would be to use the PPMI scores  $ppmi(C_j, o_i)$ . However these target values would be very different from the initial dot products, which among others means that too much of the similarity structure from the initial vector space would be lost. Instead, we will use isotonic regression to find target values  $\tau(C_j, o_i)$  for the dot product  $v_{C_j} \cdot o_i$ , which respect the ranking induced by the PPMI scores, but otherwise remain as close

<sup>1</sup>It is complete for the complexity class  $\exists\mathbb{R}$ , which sits between NP and PSPACE [29].

as possible to the initial dot products.

Let us consider a cluster  $C_j$  for which we want to determine the target feature values. Let  $o_{\sigma_1}, \dots, o_{\sigma_n}$  be an enumeration of the objects such that  $ppmi(C_j, o_{\sigma_i}) \leq ppmi(C_j, o_{\sigma_{i+1}})$  for  $i \in \{1, \dots, n-1\}$ . The corresponding target values  $\tau(C_j, o_i)$  are then obtained by solving the following optimization problem:

$$\textbf{Minimize:} \quad \sum_i (\tau(C_j, o_i) - v_{C_j} \cdot o_i)^2$$

**Subject to:**

$$\tau(C_j, o_{\sigma_1}) \leq \tau(C_j, o_{\sigma_2}) \leq \dots \leq \tau(C_j, o_{\sigma_n})$$

### 5.2.3 Fine-Tuning

We now use the target values  $\tau(C_j, o_i)$  to fine-tune the initial representations. To this end, we use a simple neural network architecture with one hidden layer. As inputs to the network, we use the initial vectors  $o_1, \dots, o_n \in \mathbb{R}^k$ . These are fed into a layer of dimension  $l$ :

$$h_i = f(Wo_i + b)$$

where  $W$  is an  $l \times k$  matrix,  $b \in \mathbb{R}^l$  is a bias term, and  $f$  is an activation function. After training the network, the vector  $h_i$  will correspond to the new representation of the  $i^{th}$  object. The vectors  $h_i$  are finally fed into an output layer containing one neuron for each cluster:

$$g_i = Dh_i$$

where  $D$  is a  $K \times l$  matrix. Note that by using a linear activation in the output layer, we can interpret the rows of the matrix  $D$  as the  $K$  feature directions, with the components of the vector  $g_i = (g_i^1, \dots, g_i^K)$  being the corresponding dot products. As the loss function for training the network, we use the squared error between the outputs  $g_i^j$  and the corresponding target values  $\tau(C_j, o_i)$ , i.e.:

$$\mathcal{L} = \sum_i \sum_j (g_i^j - \tau(C_j, o_i))^2$$

The effect of this fine-tuning step is illustrated in the right-most column of Table 5.1, where we can see that in each case the top ranked objects are now more closely related to the feature, despite being less common, and outliers such as ‘fence’ no longer appear.

| 20 Newsgroups       | F1 D1       | F1 D3       | F1 DN       |
|---------------------|-------------|-------------|-------------|
| FT MDS              | <b>0.50</b> | <b>0.47</b> | <b>0.44</b> |
| MDS                 | 0.44        | 0.42        | 0.43        |
| FT PCA              | 0.40        | 0.36        | 0.34        |
| PCA                 | 0.25        | 0.27        | 0.36        |
| FT Doc2Vec          | 0.44        | 0.42        | 0.41        |
| Doc2Vec             | 0.29        | 0.34        | 0.44        |
| FT AWV              | 0.47        | 0.45        | 0.40        |
| AWV                 | 0.41        | 0.38        | 0.43        |
| FT AWV <sub>w</sub> | 0.41        | 0.41        | 0.43        |
| AWV <sub>w</sub>    | 0.38        | 0.40        | 0.43        |
| LDA                 | 0.40        | 0.37        | 0.35        |

Table 5.2: Results for 20 Newsgroups.

## 5.3 Quantitative Evaluation

To evaluate our method, as in Chapter ?? we consider the problem of learning interpretable classifiers. In particular, we learn decision trees which are limited to depth 1 and 3, which use the rankings induced by the feature directions as input. This allows us to simultaneously assess to what extent the method can identify the right features and whether these features are modelled well using the learned directions. Note that depth 1 trees are only a single direction and a cut-off, so to perform well, the method needs to identify a highly relevant feature to the considered category. We can understand that the most demonstrable improvements for this method over the original directions will be in Depth 1 trees, as if the rankings for the important feature-directions are improved then these will be also. Depth 3 decision trees are able to model categories that can be characterized using at most three feature directions.

### Methodology

All tasks are evaluated as binary classification tasks. We randomly split the datasets into 2/3 for training and 1/3 for testing. For the place-types, we use 5-fold cross validation.

We used the logistic regression implementation from scikit-learn to find the directions.

| Movie Reviews |             |             |             |          |             |             |             |         |             |             |             |
|---------------|-------------|-------------|-------------|----------|-------------|-------------|-------------|---------|-------------|-------------|-------------|
| Genres        | D1          | D3          | DN          | Keywords | D1          | D3          | DN          | Ratings | D1          | D3          | DN          |
| FT MDS        | <b>0.57</b> | <b>0.56</b> | 0.51        | FT MDS   | <b>0.33</b> | <b>0.33</b> | 0.24        | FT MDS  | <b>0.49</b> | <b>0.51</b> | <b>0.46</b> |
| MDS           | 0.40        | 0.49        | <b>0.52</b> | MDS      | 0.31        | 0.32        | <b>0.25</b> | MDS     | 0.46        | 0.49        | <b>0.46</b> |
| FT AWV        | 0.42        | 0.42        | 0.39        | FT AWV   | 0.25        | 0.25        | 0.15        | FT AWV  | 0.47        | 0.44        | 0.39        |
| AWV           | 0.35        | 0.44        | 0.43        | AWV      | 0.26        | 0.21        | 0.19        | AWV     | 0.44        | 0.48        | 0.41        |
| LDA           | 0.52        | 0.51        | 0.45        | LDA      | 0.22        | 0.19        | 0.18        | LDA     | 0.48        | 0.48        | 0.41        |

| Place-types |             |             |             |            |             |             |             |         |             |             |             |
|-------------|-------------|-------------|-------------|------------|-------------|-------------|-------------|---------|-------------|-------------|-------------|
| Geonames    | D1          | D3          | DN          | Foursquare | D1          | D3          | DN          | OpenCYC | D1          | D3          | DN          |
| FT MDS      | 0.32        | 0.31        | 0.24        | FT MDS     | 0.41        | 0.44        | 0.41        | FT MDS  | 0.35        | 0.36        | 0.30        |
| MDS         | 0.32        | 0.31        | 0.21        | MDS        | 0.38        | 0.42        | 0.42        | MDS     | 0.35        | 0.36        | 0.29        |
| FT AWV      | 0.31        | 0.29        | 0.23        | FT AWV     | 0.39        | 0.42        | 0.41        | FT AWV  | 0.37        | <b>0.37</b> | 0.28        |
| AWV         | 0.28        | 0.28        | 0.22        | AWV        | 0.32        | 0.37        | 0.31        | AWV     | 0.33        | 0.35        | 0.26        |
| LDA         | <b>0.34</b> | <b>0.32</b> | <b>0.27</b> | LDA        | <b>0.55</b> | <b>0.48</b> | <b>0.47</b> | LDA     | <b>0.40</b> | 0.36        | <b>0.31</b> |

**Table 5.3: The results for Movie Reviews and Place-Types on depth-1, depth-3 and unbounded trees. .**

In Chapter ?? the hyper-parameters were chosen in stages. First, parameters for the best word-directions were found. Then, these best word-directions were taken and the best cluster parameters were found for these best word-directions. However, for these experimental results, we optimize the hyper-parameters together for word-directions, clustering and fine-tuning, where the best-parameters for each of these stages are those that ultimately produce the best-performing rankings for the fine-tuning on a decision tree. This is because fine-tuning is sensitive to which clusters and directions are included, as optimizing the ranking for one feature-direction may disrupt the ranking for another. This can be illustrated by the idea of optimizing a ranking for a direction on a noisy term like 'berardin', which refers to some metadata from the review text was optimized, then it's unlikely that this would benefit the other directions. However, if multiple directions that correspond to different genres were optimized like 'Horror' and 'Funny', then

| IMDB Sentiment | D1          | D3          | DN          |
|----------------|-------------|-------------|-------------|
| FT PCA         | 0.78        | 0.80        | 0.79        |
| PCA            | 0.76        | <b>0.82</b> | <b>0.80</b> |
| FT AWV         | 0.72        | 0.76        | 0.71        |
| AWV            | 0.74        | 0.76        | 0.71        |
| LDA            | <b>0.79</b> | 0.80        | 0.79        |

Table 5.4: Results for IMDB Sentiment.

it's likely that they would all benefit from a better representation. Cluster-directions are used because if all hyper-parameters are trained together, we can expect to find a set of directions that work with each other more easily than by limiting frequency for word-directions.

We evaluate for all domains described in Chapter 3 excluding Reuters. When learning word directions, only sufficiently frequent words are considered. In Chapter ?? this was chosen as a hyper-parameter, but as all parameters for each stage are tuned together it would take far too much time to optimize in this way, so it is chosen beforehand. It is chosen by pre-determining thresholds loosely based on the size of the vocabulary for the domain. We chose 100 for the movies dataset, 50 for the place-types, 30 for the 20 newsgroups dataset, and 50 for the IMDB sentiment dataset.

For hyperparameter tuning, we take 20% of the data from the training split as development data. We choose the hyperparameter values that maximize the F1 score on the development data for a Decision Tree on the improved feature-rankings that the fine-tuning network produces. As candidate values for the number of dimensions of the vector spaces we used  $\{50, 100, 200\}$ . The number of directions to be used as input to the clustering algorithm was chosen from  $\{500, 1000, 2000\}$ . The number of clusters was chosen from  $\{k, 2k\}$ , with  $k$  the chosen number of dimensions. For the hidden layer of the neural network, we fixed the number of dimensions as equal to the number of clusters. As the scoring metric for the dimensions, we considered accuracy, Kappa and NDCG. In all experiments, we used 300 epochs, a minibatch size of 200, and the tanh activation function for the hidden layer of the neural network. After some preliminary tests we found that in most cases the parameters for the network could be kept the same. In all experiments: 300 epochs, batch size 200 and tanh activation for the hidden layer. The hidden layer was kept the same size as the input space  $V_n$ . We train the network using AdaGrad [7],



with default values, and the model was implemented in the Keras library.

For the cluster size, we follow work by Steven Schockaert[29] and use twice the amount of clusters as there are dimensions in the space.

To learn the decision trees, we use the scikit-learn implementation of CART, which allows us to limit the depth of the trees. setting the maximum depth to one, three, or not at all. We used information gain as the attribute selection criterion. To mitigate the effects of class imbalance, the less frequent class was given a higher weight during training.

### 5.3.1 Results

Table 5.2 shows the results for the 20 newsgroups dataset, where we use FT to indicate the results with fine-tuning<sup>2</sup>. We can see that the fine-tuning method consistently improves the performance of the depth-1 and depth-3 trees, often in a very substantial way. After fine-tuning, the results are also consistently better than those of LDA. For the unbounded trees (DN), the differences are small and fine-tuning sometimes even makes the results worse. This can be explained by the fact that the fine-tuning method specializes the space towards the selected features, which means that some of the structure of the initial space will be distorted. Unbounded decision trees are far less sensitive to the quality of the directions, and can even perform reasonably on random directions. Interestingly, depth-1 trees achieved the best overall performance, with depth-3 trees and especially unbounded trees overfitting. Since MDS and AWW perform best, we have only considered these two representations (along with LDA) for the remaining datasets, except for the IMDB Sentiment dataset, which is too large for using MDS.

The results for the movies and place-types datasets are shown in Table 5.3. For the MDS representations, the fine-tuning method again consistently improved the results for D1 and D3 trees. For the AWW representations, the fine-tuning method was also effective in most cases, although there are a few exceptions. What is noticeable is that for movie genres, the improvement is substantial, which reflects the fact that genres are a salient property of movies. For example, the decision tree for the genre ‘Horror’ could use the feature direction for  $\{gore, gory, horror, gruesome\}$ .

<sup>2</sup>Since the main purpose of this first experiment was to see whether fine-tuning improved consistently across a broad set of representations, here we considered a slightly reduced pool of parameter values for hyperparameter tuning.

Some of the other datasets refer to more specialized properties, and the performance of our method then depends on whether it has identified features that relate to these properties. It can be expected that a supervised variant of this method would perform consistently better in such cases. After fine-tuning, the MDS based representation outperforms LDA on the movies dataset, but not for the place-types. This is a consequence of the fact that some of the place-type categories refer to very particular properties, such as geological phenomena, which may not be particularly dominant among the Flickr tags that were used to generate the spaces. In such cases, using a BoW based representation may be more suitable.

The results for IMDB Sentiment are shown in Table 5.4. In this case, the fine-tuning method fails to make meaningful improvements, and in some cases actually leads to worse results. This can be explained from the fact that the feature directions which were found for this space are themes and properties, rather than aspects of binary sentiment evaluation. The fine-tuning method aims to improve the representation of these properties, possibly at the expense of other aspects.

## 5.4 Conclusions

We have introduced a method to identify and model the salient features from a given domain as directions in a semantic space. Our method is based on the observation that there is a trade-off between accurately modelling similarity in a vector space, and faithfully modelling features as directions. In particular, we introduced a post-processing step, modifying the initial semantic space, which allows us to find higher-quality directions. We provided qualitative examples that illustrate the effect of this fine-tuning step, and quantitatively evaluated its performance in a number of different domains, and for different types of semantic space representations. We found that after fine-tuning, the feature directions model the objects in a more meaningful way. This was shown in terms of an improved performance of low-depth decision trees in natural categorization tasks. However, we also found that when the considered categories are too specialized, the fine-tuning method was less effective, and in some cases even led to a slight deterioration of the results. We speculate that performance could be improved for such categories by integrating domain knowledge into the fine-tuning method.

---

## ***Chapter 6***

# **Investigating Neural Networks In Terms Of Directions**

## **6.1 Chapter 5**

Neural network models that encode spatial relationships in their hidden layers have achieved state-of-the-art in Text Classification by using transfer learning from a pre-trained Language Model [8]. There have also been neural network models that produce an interpretable representation, for example InfoGan. Most state-of-the-art results rely on Vector Space Models. Ideally the method would be able to achieve strong results for simple interpretable classifiers by transforming an existing representation that performs well at the task.

### **6.1.1 Chapter 3 Space Types**

| Genres     |                   | Keywords  |         |         | Ratings  |         |         |
|------------|-------------------|-----------|---------|---------|----------|---------|---------|
| Movies     | D1                | D2        | D3      | D1      | D2       | D3      |         |
|            | 50 PCA            | 50 MDS    | 100 MDS | 200 PCA | 200 MDS  | 200 PCA | 50 PCA  |
|            | Single directions | N/A       | N/A     | N/A     | N/A      | N/A     | N/A     |
| Newsgroups |                   | Sentiment |         |         | Reuters  |         |         |
| Rep        | 200 PCA           | 200 PCA   | 100 PCA | PCA 100 | PCA 50   | 200 PCA | 100 PCA |
| Single dir | 200 MDS           | 100 D2V   | 50 D2V  | D2V 100 | PCA 50   | N/A     | N/A     |
| Foursquare |                   | OpenCYC   |         |         | Geonames |         |         |
| Placetypes | D1                | D2        | D3      | D1      | D2       | D3      |         |
| Rep        | MDS 100           | AWV 50    | MDS 200 | AWV 50  | MDS 200  | MDS 50  | AWV 200 |
| Single dir | N/A               | N/A       | N/A     | N/A     | N/A      | N/A     | N/A     |

**Table 6.1: Space-types, clusters have the same as single directions.**

---

## ***Chapter 7***

# **Appendix**

## **7.1 Chapter 3**

### **7.1.1 Difference between Representations and Single Directions**

| Newsgroups | D1           |              |              |              | D2           |              |              |              | D3           |              |              |              | D3           |    |      |     |
|------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|----|------|-----|
|            | ACC          | F1           | Prec         | Rec          | ACC          | F1           | Prec         | Rec          | ACC          | F1           | Prec         | Rec          | ACC          | F1 | Prec | Rec |
| PCA 200    | 0.254        | 0.097        | 0.373        | -0.55        | 0.117        | 0.058        | 0.433        | -0.41        | 0.004        | 0.099        | 0.134        | 0.078        |              |    |      |     |
| PCA 100    | 0.259        | 0.135        | 0.345        | -0.5         | 0.126        | 0.112        | 0.438        | -0.367       | 0.006        | 0.157        | 0.118        | <b>0.149</b> |              |    |      |     |
| PCA 50     | 0.277        | 0.133        | 0.277        | -0.492       | 0.129        | 0.123        | 0.387        | -0.347       | 0.006        | 0.177        | 0.228        | 0.143        |              |    |      |     |
| AWV 200    | 0.145        | 0.133        | 0.1          | -0.005       | 0.199        | 0.128        | 0.362        | -0.414       | 0.194        | 0.185        | 0.441        | -0.397       |              |    |      |     |
| AWV 100    | 0.153        | 0.133        | 0.098        | 0.01         | 0.043        | 0.084        | 0.06         | <b>0.109</b> | 0.21         | 0.137        | 0.414        | -0.474       |              |    |      |     |
| AWV 50     | 0.11         | 0.122        | 0.088        | 0.044        | 0.056        | 0.088        | 0.068        | 0.052        | 0.21         | 0.142        | 0.362        | -0.468       |              |    |      |     |
| MDS 200    | <b>0.378</b> | <b>0.234</b> | <b>0.439</b> | -0.498       | <b>0.22</b>  | 0.203        | 0.509        | -0.372       | 0.243        | <b>0.257</b> | <b>0.568</b> | -0.406       |              |    |      |     |
| MDS 100    | 0.271        | 0.178        | 0.138        | -0.108       | 0.205        | 0.167        | 0.465        | -0.401       | <b>0.254</b> | 0.217        | 0.506        | -0.459       |              |    |      |     |
| MDS 50     | 0.228        | 0.171        | 0.119        | <b>0.115</b> | 0.126        | 0.136        | 0.108        | 0.014        | 0.222        | 0.155        | 0.452        | -0.476       |              |    |      |     |
| D2V 200    | 0.149        | 0.138        | 0.101        | 0.037        | 0.158        | 0.202        | <b>0.514</b> | -0.288       | 0.192        | 0.225        | 0.526        | -0.365       |              |    |      |     |
| D2V 100    | 0.162        | 0.166        | 0.123        | 0.041        | 0.169        | <b>0.222</b> | 0.474        | -0.266       | 0.176        | 0.249        | 0.505        | -0.306       |              |    |      |     |
| D2V 50     | 0.162        | 0.181        | 0.132        | 0.08         | 0.154        | 0.193        | 0.452        | -0.299       | 0.181        | 0.256        | 0.501        | -0.314       |              |    |      |     |
|            |              |              |              |              |              |              |              |              |              |              |              |              |              |    |      |     |
| Reuters    | D1           |              |              |              | D2           |              |              |              | D3           |              |              |              | Sentiment    |    |      |     |
|            | ACC          | F1           | ACC          | F1           | ACC          | F1           | Prec         | Rec          | ACC          | F1           | ACC          | F1           | ACC          | F1 | ACC  | F1  |
| PCA        | 0.129        | 0.33         | 0.062        | 0.265        | -0.002       | -0.034       | PCA          | -0.006       | 0.053        | 0.042        | 0.044        | 0.024        | 0.032        |    |      |     |
| AWV        | <b>0.193</b> | 0.345        | 0.008        | 0.327        | <b>0.007</b> | <b>0.243</b> | AWV          | 0.057        | 0.047        | 0.068        | 0.042        | 0.028        | 0.018        |    |      |     |
| MDS        | 0.184        | <b>0.414</b> | 0.08         | 0.349        | 0.003        | 0.231        | D2V          | <b>0.134</b> | <b>0.12</b>  | <b>0.122</b> | <b>0.094</b> | <b>0.12</b>  | <b>0.121</b> |    |      |     |
| D2V        | 0.159        | 0.316        | <b>0.112</b> | <b>0.366</b> | 0.006        | 0.188        |              |              |              |              |              |              |              |    |      |     |
|            |              |              |              |              |              |              |              |              |              |              |              |              |              |    |      |     |
| Placetypes | D1           |              |              |              | D2           |              |              |              | D3           |              |              |              | Movies       |    |      |     |
|            | ACC          | F1           | ACC          | F1           | ACC          | F1           | ACC          | F1           | ACC          | F1           | ACC          | F1           | ACC          | F1 | ACC  | F1  |
| OpenCYC    |              |              |              |              |              |              | Genres       |              |              |              |              |              |              |    |      |     |
| PCA        | <b>0.047</b> | <b>0.025</b> | -0.003       | 0.038        | <b>0.04</b>  | <b>0.024</b> | PCA          | 0.102        | 0.111        | <b>0.064</b> | 0.101        | <b>0.196</b> | <b>0.142</b> |    |      |     |
| AWV        | 0.036        | 0.021        | <b>0.083</b> | <b>0.052</b> | 0.027        | -0.006       | AWV          | 0.132        | 0.132        | 0.064        | <b>0.115</b> | 0.156        | 0.114        |    |      |     |
| MDS        | 0.034        | 0.009        | 0.011        | -0.012       | 0.016        | -0.024       | MDS          | <b>0.17</b>  | <b>0.148</b> | 0.049        | 0.104        | 0.145        | 0.141        |    |      |     |
|            |              |              |              |              |              |              |              |              |              |              |              |              |              |    |      |     |
| Foursquare | D1           |              |              |              | D2           |              |              |              | D3           |              |              |              | Keywords     |    |      |     |
|            | ACC          | F1           | ACC          | F1           | ACC          | F1           | ACC          | F1           | ACC          | F1           | ACC          | F1           | ACC          | F1 | ACC  | F1  |
| PCA        | 0.054        | 0.135        | <b>0.084</b> | <b>0.082</b> | 0.008        | <b>0.143</b> | PCA          | 0.09         | 0.04         | <b>0.083</b> | 0.034        | 0.032        | 0.022        |    |      |     |
| AWV        | <b>0.151</b> | <b>0.14</b>  | 0.053        | 0.02         | <b>0.038</b> | 0.014        | AWV          | <b>0.156</b> | 0.041        | 0.031        | 0.024        | <b>0.057</b> | <b>0.025</b> |    |      |     |
| MDS        | -0.094       | -0.022       | 0.075        | 0.055        | 0.038        | 0.031        | MDS          | 0.111        | <b>0.051</b> | 0.05         | <b>0.035</b> | 0.033        | 0.023        |    |      |     |
|            |              |              |              |              |              |              |              |              |              |              |              |              |              |    |      |     |
| Geonames   | D1           |              |              |              | D2           |              |              |              | D3           |              |              |              | Ratings      |    |      |     |
|            | ACC          | F1           | ACC          | F1           | ACC          | F1           | ACC          | F1           | ACC          | F1           | ACC          | F1           | ACC          | F1 | ACC  | F1  |
| PCA        | <b>0.163</b> | 0.047        | 0.063        | 0.037        | <b>0.063</b> | 0.011        | PCA          | -0.003       | 0.003        | 0.04         | 0.023        | -0.003       | 0.007        |    |      |     |
| AWV        | 0.054        | <b>0.119</b> | 0.04         | <b>0.21</b>  | -0.039       | <b>0.106</b> | AWV          | <b>0.045</b> | <b>0.041</b> | <b>0.074</b> | <b>0.042</b> | <b>0.08</b>  | 0.036        |    |      |     |
| MDS        | -0.035       | -0.06        | <b>0.078</b> | -0.007       | -0.032       | 0.08         | MDS          | 0.028        | 0.026        | 0.057        | 0.04         | 0.055        | <b>0.045</b> |    |      |     |

Table 7.1: The difference between the representations being directly input to the low-depth decision trees and the word directions

### **7.1.2 Class Names and Positive Occurrences**

| Newsgroups                        | Positives | OpenCYC                      | Positives | FourSquare                 | Positives | Geonames                     | Positives | Genres                    | Positives | Ratings      | Positives |
|-----------------------------------|-----------|------------------------------|-----------|----------------------------|-----------|------------------------------|-----------|---------------------------|-----------|--------------|-----------|
| alt.atheism                       | 799       | aqueduct                     | 67        | ArtsAndEntertainment       | 39        | StreamLake                   | 74        | Action                    | 2105      | USA-G        | 1974      |
| comp.graphics                     | 973       | border                       | 556       | CollegeAndUniversity       | 33        | ParksArea                    | 28        | Adventure                 | 1451      | UK-12-12A    | 1566      |
| comp.os.ms-windows.misc           | 985       | building                     | 91        | Food                       | 82        | RoadRailroad                 | 16        | Animation                 | 396       | UK-15        | 3957      |
| comp.sys.ibm.pc.hardware          | 982       | dam                          | 389       | ProfessionalAndOtherPlaces | 47        | SpotBuildingFarm             | 176       | Biography                 | 627       | UK-18        | 2009      |
| comp.sys.mac.hardware             | 963       | facility                     | 173       | NightlifeSpot              | 17        | MountainHillRock             | 68        | Comedy                    | 4566      | UK-PG        | 1724      |
| comp.windows.x                    | 988       | foreground                   | 43        | ParksAndOutdoors           | 44        | Undersea                     | 27        | Crime                     | 2073      | USA-PG-PG13  | 439       |
| misc.forsale                      | 975       | historical_site              | 297       | ShopsAndService            | 88        | ForestHeath                  | 14        | Documentary               | 781       | USA-R        | 5170      |
| rec.autos                         | 990       | holy_site                    | 44        | TravelAndTransport         | 35        |                              |           | Drama                     | 7269      |              |           |
| rec.motorcycles                   | 996       | landmark                     | 96        | Residence                  | 6         |                              |           | Family                    | 873       |              |           |
| rec.sport.baseball                | 994       | medical_facility             | 28        |                            |           |                              |           | Fantasy                   | 928       |              |           |
| rec.sport.hockey                  | 999       | medical_school               | 49        |                            |           |                              |           | Film-Noir                 | 170       |              |           |
| sci.crypt                         | 991       | military_place               | 30        |                            |           |                              |           | History                   | 502       |              |           |
| sci.electronics                   | 984       | monsoon_forest               | 53        |                            |           |                              |           | Horror                    | 1963      |              |           |
| sci.med                           | 990       | national_monument            | 145       |                            |           |                              |           | Music                     | 1051      |              |           |
| sci.space                         | 987       | outdoor_location             | 103       |                            |           |                              |           | Musical                   | 529       |              |           |
| soc.religion.christian            | 997       | rock_formation               | 184       |                            |           |                              |           | Mystery                   | 1128      |              |           |
| talk.politics.guns                | 910       | room                         | 60        |                            |           |                              |           | Romance                   | 2965      |              |           |
| talk.politics.mideast             | 940       |                              |           |                            |           |                              |           | Sci-Fi                    | 1266      |              |           |
| talk.politics.misc                | 775       |                              |           |                            |           |                              |           | Short                     | 560       |              |           |
| talk.religion.misc                | 628       |                              |           |                            |           |                              |           | Sport                     | 385       |              |           |
|                                   |           |                              |           |                            |           |                              |           | Thriller                  | 3293      |              |           |
|                                   |           |                              |           |                            |           |                              |           | War                       | 671       |              |           |
|                                   |           |                              |           |                            |           |                              |           | Western                   | 454       |              |           |
| Keywords (1)                      | Positives | Keywords (2)                 | Positives | Keywords (3)               | Positives | Keywords (4)                 | Positives | Keywords (5)              | Positives | Reuters      | Positives |
| adultery                          | 853       | dancing                      | 1655      | funeral                    | 802       | money                        | 887       | shot-to-death             | 976       | trade        | 466       |
| bar                               | 1334      | death                        | 2596      | gore                       | 820       | mother-daughter-relationship | 1477      | singer                    | 1278      | grain        | 580       |
| bare-breasts                      | 1360      | doctor                       | 1193      | gun                        | 1445      | mother-son-relationship      | 1908      | singing                   | 1372      | nat-gas      | 105       |
| bare-chested-male                 | 1360      | dog                          | 1605      | gunfight                   | 776       | murder                       | 3496      | song                      | 986       | crude        | 568       |
| based-on-novel                    | 2390      | drink                        | 1080      | helicopter                 | 864       | new-york-city                | 1464      | suicide                   | 1092      | sugar        | 162       |
| beach                             | 881       | drinking                     | 1246      | hero                       | 789       | nudity                       | 1887      | surprise-ending           | 1202      | corn         | 237       |
| beating                           | 1011      | drunkmess                    | 1291      | horse                      | 825       | one-word-title               | 1357      | tears                     | 892       | veg-oil      | 124       |
| betrayal                          | 848       | escape                       | 789       | hospital                   | 1434      | party                        | 1131      | telephone-call            | 1187      | ship         | 280       |
| blood                             | 2384      | explosion                    | 1283      | hotel                      | 902       | photograph                   | 1304      | title-spoken-by-character | 1725      | coffee       | 139       |
| boy                               | 824       | face-slap                    | 907       | husband-wife-relationship  | 2392      | pistol                       | 1378      | topless-female-nudity     | 1079      | wheat        | 283       |
| boyfriend-girlfriend-relationship | 1093      | falling-from-height          | 875       | independent-film           | 3431      | police                       | 1801      | train                     | 1069      | gold         | 120       |
| brother-brother-relationship      | 884       | family-relationships         | 1787      | infidelity                 | 862       | policeman                    | 792       | underwear                 | 860       | acq          | 2363      |
| brother-sister-relationship       | 1025      | father-daughter-relationship | 1758      | jealousy                   | 928       | pregnancy                    | 821       | violence                  | 2231      | interest     | 457       |
| character-name-in-title           | 2146      | father-son-relationship      | 2201      | kidnapping                 | 863       | punched-in-the-face          | 870       | voice-over-narration      | 1058      | money-fx     | 676       |
| chase                             | 1351      | female-nudity                | 2328      | kiss                       | 1759      | rain                         | 1053      | watching-tv               | 887       | soybean      | 111       |
| church                            | 897       | fight                        | 1356      | knife                      | 1097      | restaurant                   | 1202      | wedding                   | 800       | oilseed      | 171       |
| cigarette-smoking                 | 1858      | fire                         | 1027      | love                       | 2164      | revenge                      | 1336      | earn                      | 3951      | earn         | 3951      |
| corpse                            | 1008      | fistfight                    | 977       | machine-gun                | 878       | sequel                       | 801       | bop                       | 104       | bop          | 104       |
| crying                            | 1149      | flashback                    | 1937      | male-nudity                | 1122      | sex                          | 2126      | gnp                       | 136       | gnp          | 136       |
| cult-film                         | 1636      | friend                       | 1193      | marriage                   | 1407      | shootout                     | 1174      | dlr                       | 162       | dlr          | 162       |
| dancer                            | 1020      | friendship                   | 1903      | martial-arts               | 824       | shot-in-the-chest            | 892       | money-supply              | 168       | money-supply | 168       |

Table 7.2: Positive Instance Counts for each Class



# GNU Free Documentation License

Version 1.2, November 2002

Copyright © 2000, 2001, 2002 Free Software Foundation, Inc.  
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

## 0. Preamble

The purpose of this License is to make a manual, textbook, or other functional and useful document *free* in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of “copyleft”, which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

## 1. Applicability and Definitions

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The “Document”, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as “you”. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A “Modified Version” of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A “Secondary Section” is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document’s overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The “Invariant Sections” are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The “Cover Texts” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A “Transparent” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not “Transparent” is called “Opaque”.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format,  $\LaTeX$  input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The “Title Page” means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, “Title Page” means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text.

A section “Entitled XYZ” means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as “Acknowledgements”, “Dedications”, “Endorsements”, or “History”.) To “Preserve the Title” of such a section when you modify the Document means that it remains a section “Entitled XYZ” according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

## 2. Verbatim Copying

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

## 3. Copying in Quantity

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

## 4. Modifications

you may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

- K.** For any section Entitled “Acknowledgements” or “Dedications”, Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L.** Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M.** Delete any section Entitled “Endorsements”. Such a section may not be included in the Modified Version.
- N.** Do not retitle any existing section to be Entitled “Endorsements” or to conflict in title with any Invariant Section.
- O.** Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version’s license notice. These titles must be distinct from any other section titles.

You may add a section Entitled “Endorsements”, provided it contains nothing but endorsements of your Modified Version by various parties — for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

## 5. Combining Documents

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known,

or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled “History” in the various original documents, forming one section Entitled “History”; likewise combine any sections Entitled “Acknowledgements”, and any sections Entitled “Dedications”. You must delete all sections Entitled “Endorsements.”

## 6. Collections of Documents

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

## 7. Aggregation with Independent Works

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an “aggregate” if the copyright resulting from the compilation is not used to limit the legal rights of the compilation’s users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document’s Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

## 8. Translation

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between

the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled “Acknowledgements”, “Dedications”, or “History”, the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

## 9. Termination

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

## 10. Future Revisions of this License

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License “or any later version” applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

## ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright © YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled “GNU Free Documentation License”.

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the “with... Texts.” line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.



# Bibliography

- [1] David Arthur and Sergei Vassilvitskii. k-means ++ : The Advantages of Careful Seeding. 8:1–11.
- [2] Andrew M. Dai, Christopher Olah, and Quoc V. Le. Document embedding with paragraph vectors. *CoRR*, abs/1507.07998, 2015.
- [3] Rajarshi Das, Manzil Zaheer, and Chris Dyer. Gaussian LDA for topic models with word embeddings. In *Proc. ACL*, pages 795–804, 2015.
- [4] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.
- [5] J. Derrac and S. Schockaert. Inducing semantic relations from conceptual spaces: a data-driven approach to plausible reasoning. *Artificial Intelligence*, pages 74–105, 2015.
- [6] Joaquin Derrac and Steven Schockaert. Inducing semantic relations from conceptual spaces: A data-driven approach to plausible reasoning. *Artificial Intelligence*, 228:66–94, 2015.
- [7] John Duchi, Elad Hazan, and Yoram Singer. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *Journal of Machine Learning Research*, 12:2121–2159, 2011.
- [8] Chengyue Gong. FRAGE : Frequency-Agnostic Word Representation. 1(Nips):1–15, 2018.
- [9] Abhijeet Gupta, Gemma Boleda, Marco Baroni, and Sebastian Padã<sup>3</sup>. *Distributional vectors encode referential attributes*. In *Proceedings of the 2015 Conference on*
- [10] Felix Hill, Kyunghyun Cho, and Anna Korhonen. Learning distributed representations of sentences from unlabelled data. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1367–1377, 2016.

- [11] Johan Huysmans, Karel Dejaeger, Christophe Mues, Jan Vanthienen, and Bart Baesens. An empirical evaluation of the comprehensibility of decision table, tree and rule based predictive models. *Decision Support Systems*, 51(1):141–154, 2011.
- [12] Shoaib Jameel, Zied Bouraoui, and Steven Schockaert. MEmBER : Max-Margin Based Embeddings for Entity Retrieval.
- [13] Shoaib Jameel, Zied Bouraoui, and Steven Schockaert. Member: Max-margin based embeddings for entity retrieval. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 783–792, 2017.
- [14] Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems*, 20(4):422–446, 2002.
- [15] Joo-kyung Kim. Deriving adjectival scales from continuous space word representations. (October):1625–1630, 2013.
- [16] Adriana Kovashka, Devi Parikh, and Kristen Grauman. WhittleSearch : Image Search with Relative Attribute Feedback.
- [17] Adriana Kovashka, Devi Parikh, and Kristen Grauman. Whittlesearch: Image search with relative attribute feedback. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2973–2980, 2012.
- [18] Quoc V. Le and Tomas Mikolov. Distributed representations of sentences and documents. In *Proceedings of the 31th International Conference on Machine Learning*, pages 1188–1196, 2014.
- [19] Yang Liu, Zhiyuan Liu, Tat-Seng Chua, and Maosong Sun. Topical word embeddings. In *Proc. AAAI*, pages 2418–2424, 2015.
- [20] Andrew L Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. Learning Word Vectors for Sentiment Analysis. pages 142–150, 2011.
- [21] Junhua Mao, Wei Xu, Yi Yang, Jiang Wang, and Alan L. Yuille. Explain Images with Multimodal Recurrent Neural Networks. *arXiv:1410.1090 [cs]*, pages 1–9, 2014.
- [22] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2, NIPS’13*, pages 3111–3119, USA, 2013. Curran Associates Inc.
- [23] Menaka Narayanan, Emily Chen, Jeffrey He, Been Kim, Sam Gershman, and Finale Doshi-Velez. How do Humans Understand Explanations from Machine Learning Systems? An Evaluation of the Human-Interpretability of Explanation. pages 1–21, 2018.

- [24] Hamid Palangi, Paul Smolensky, Xiaodong He, and Li Deng. Question-Answering with Grammatically-Interpretable Representations. 2017.
- [25] 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. Association for Computational Linguistics, 2014.
- [26] Jeffrey Pennington, Richard Socher, and Christopher D Manning. GloVe: Global Vectors for Word Representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1532–1543, 2014.
- [27] Sascha Rothe and Language Processing. Word Embedding Calculus in Meaningful Ultradense Subspaces. pages 512–517, 2016.
- [28] T.L. Saaty and M.S. Ozdemir. Why the magic number seven plus or minus two. *Mathematical and Computer Modelling*, 38(3):233–244, 2003.
- [29] Steven Schockaert and Jae Hee Lee. Qualitative reasoning about directions in semantic spaces. In Qiang Yang and Michael Wooldridge, editors, *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence*, pages 3207–3213. AAAI Press, 2015.
- [30] D Sculley. Web-Scale K-Means Clustering. pages 4–5, 2010.
- [31] Christophe Van Gysel, Maarten de Rijke, and Evangelos Kanoulas. Learning latent vector spaces for product search. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 165–174, 2016.
- [32] Paolo Viappiani. Preference-based Search using Example-Critiquing with Suggestions. 27:465–503, 2006.
- [33] Paolo Viappiani, Boi Faltings, and Pearl Pu. Preference-based search using example-critiquing with suggestions. *Journal of Artificial Intelligence Research*, 27:465–503, 2006.
- [34] Jesse Vig, Shilad Sen, and John Riedl. The tag genome: Encoding community knowledge to support novel interaction. *ACM Transactions on Interactive Intelligent Systems*, 2(3):13:1–13:44, 2012.
- [35] Jesse Vig, Shilad Sen, and John Riedl. The Tag Genome : Encoding Community Knowledge to Support Novel Interaction The Tag Genome : Encoding Community Knowledge to Support. (November), 2014.
- [36] Youwei Zhang and Laurent El Ghaoui. Large-Scale Sparse Principal Component Analysis with Application to Text Data. pages 1–8, 2012.