Data & Knowledge Engineering 121 (2019) 88–108

Contents lists available at ScienceDirect

Data & Knowledge Engineering

journal homepage: www.elsevier.com/locate/datak

A Hybrid Semantic Knowledgebase-Machine Learning Approach for Opinion Mining

Rowida Alfrjani, Taha Osman *, Georgina Cosma

Department of Computing, Nottingham Trent University, Nottingham, UK

ARTICLE INFO

Keywords:
Feature extraction
Classification
Semantic web
Knowledgebase

ABSTRACT

Opinion mining tools enable users to efficiently process a large number of online reviews in order to determine the underlying opinions. This paper presents a Hybrid Semantic Knowledgebase-Machine Learning approach for mining opinions at the domain feature level and classifying the overall opinion on a multi-point scale. The proposed approach benefits from the advantages of deploying a novel Semantic Knowledgebase approach to analyse a collection of reviews at the domain feature level and produce a set of structured information that associates the expressed opinions with specific domain features. The information in the knowledgebase is further supplemented with domain-relevant facts sourced from public Semantic datasets, and the enriched semantically-tagged information is then used to infer valuable semantic information about the domain as well as the expressed opinions on the domain features by summarising the overall opinions about the domain across multiple reviews, and by averaging the overall opinions about other cinematic features. The retrieved semantic information represents a valuable resource for modelling a machine learning classifier to predict the numerical rating of each review. Experimental evaluation revealed that the proposed Hybrid Semantic Knowledgebase-Machine Learning approach improved the precision and recall of the extracted domain features, and hence proved suitable for producing an enriched dataset of semantic features that resulted in higher classification accuracy.

1. Introduction

Consumers often consult the opinion of others when considering purchasing decisions. For instance, it is common to seek out the opinion of friends, favourite bloggers and reviewers to make a decision about purchasing a product, voting for a political candidate or choosing a movie. Opinions, often in the form of reviews, are increasingly being published on websites, blogs and social media outlets. Organisations invest considerable resources to collate and analyse online material in order to analyse the underlying user trends regarding consumer sentiments, and use such information to improve their products and services and to shape their production strategies and marketing campaigns. The challenge is that online opinions are predominantly expressed in natural language text, and hence opinion mining tools are required to facilitate the effective extraction and analysis of opinions from unstructured text. Such tools often adopt algorithms from Natural Language Processing, Information Retrieval and Machine Learning disciplines.

Opinion mining is commonly implemented by extracting contents for a specific domain (e.g. movie, music, and product) and performing opinion mining at various levels of text granularity: document, sentence or domain feature level. At document and sentence level, opinion mining aims to classify the overall sentiment orientation that is expressed in a document [1], [2], [3] or a sentence [4], [5]. Opinion mining at the domain feature level is considered to be a difficult task because it requires deep understanding of the sentence structure and knowledge of the problem domain (e.g. movie reviews) in order to correctly classify

* Corresponding author.
E-mail address: taha.osman@ntu.ac.uk (T. Osman).

https://doi.org/10.1016/j.datak.2019.05.002
Received 26 January 2018; Received in revised form 11 March 2019; Accepted 10 May 2019
Available online 20 May 2019

sentences based on their polarity. Particularly challenging is the extraction of the domain feature mentions (e.g. actress, show, script, story) from the reviews and associating each domain feature with its corresponding sentiment to determine its polarity score (e.g. the beauty of the script +1; Bulletproof Heart is not an excellent movie −1; The great Matt Craven will probably be forever remembered +1). Opinion mining at domain feature level can be further considered for enhancing the opinion classification task via summing or averaging the polarity score of each extracted domain feature to determine the numerical rating of the review (e.g. 4,3,2,1 and 0 for very positive, positive, neutral, negative, and very negative respectively).

Opinion mining research at domain feature level employs different approaches such as Association Rule Mining, Machine Learning and Semantic Knowledgebase approaches to primarily improve the outcome of the domain feature extraction task, which consequently enhances the performance of opinion classification task. Association Rule Mining approaches primarily rely on Natural Language Processing techniques to identify nouns and noun phrases to be domain features, whereas Machine Learning approaches rely on a large set of training data to learn domain features from reviews. More recently, Semantic Knowledgebase approaches have been deployed for domain feature extraction with promising success. The Semantic Knowledgebase approaches are based on utilising domain knowledgebase to extract domain features from reviews, which contains a conceptualised knowledge background of the domain. The Domain Knowledge captures the key concepts and relations of the problem domain's environment, which is then populated with entities and facts/events that subscribe to the modelled concepts and relations [6]. Such domain knowledge can be utilised to improve the performance of domain feature extraction task. Semantic Web technologies organise knowledge in a formalised Semantic Knowledgebase that provides efficient support for linking and sharing data between resources, and presenting data in a way that computer machines can process. In addition, the formalised Semantic Knowledgebase is capable of presenting the domain knowledge in a structured and consistent manner which facilitates the qualitative interpretation of domain specific contents in a way that people can understand. Moreover, Semantic Web technologies provide support for enriching the modelled domain knowledgebase with relevant ground facts from public-sourced Linked Open Data resources. A chief challenge for the Semantic Knowledgebase approaches is the collection of sufficient domain-related data to produce a sufficiently rich Semantic Knowledgebase that drives the domain feature extraction task [7].

Recently, the integration of Semantic knowledgebase and Machine Learning methods has emerged as an effective approach for improving the process of opinion mining at domain feature level, in particular, for enhancing the opinion classification task [8], [9]. However, the reported efforts have mainly focused on binary classification tasks, i.e., identifying whether the content has a positive or negative opinion. The work presented in this paper is principally motivated by the need to develop a Hybrid Semantic Knowledgebase-Machine Learning approach to enhance the performance of opinion mining at domain feature level and to improve opinion classification on a multi-point scale (i.e., multi-class classification). The proposed approach has been applied to the movie reviews problem domain.

The remainder of the paper is organised as follows: research contribution is presented in Section 2 and related work is discussed in Section 3. Section 4 discusses in detail the proposed Hybrid Semantic Knowledgebase-Machine Learning approach for opinion mining, which covers capturing domain knowledgebase, extracting and classification processes. Section 5 discusses the experimental evaluation, and Section 6 provides an overview of the proposed approach and how it can be applied to other domains. Finally, a conclusion and discussion of future work are presented in Section 7.

2. Contribution

The contribution of the proposed Hybrid Semantic Knowledgebase-Machine Learning approach can be summarised as follows:

(1) A new Domain Feature Extraction algorithm that improves the precision and recall of the extracted domain features. The Domain Feature Extraction algorithm utilises a comprehensive knowledge of the chosen domain (key concepts and their synonyms and ground facts) and public Linked Open Data sources such as DBpedia and Internet Movie Database (IMDb).
(2) A new Domain Feature-Sentiment Association algorithm that removes false positive opinions (i.e., the domain feature-sentiment pairs) that objectively describe factual information using a generated sentiment lexicon for each domain feature.
(3) A new Opinion Classification algorithm that delivers enhanced opinion classification on a multi-point scale. The Opinion Classification algorithm generates an enriched set of semantic data from a modelled Semantic Knowledgebase and merge it with a statistical dataset and then use that as input into machine learning algorithms. The Opinion Classification algorithm appears to be the first study that presents such kind of combination between Semantic Knowledgebase and Machine Learning approaches for classifying opinions on a multi-point scale.

3. Related work

This section discusses related literature in opinion mining with a focus on methods for extracting domain features from natural language text reviews, and methods for classifying opinions.

3.1. Domain feature extraction

The Association Rule Mining approach, which primarily relies on Natural Language Processing techniques, is the most popular for domain feature extraction. Hu and Liu [10] extracted frequent nouns or noun-phrases to be domain features using an Apriori algorithm, whereas the approach by Eirinaki et al. [11] involved initially extracting nouns, and then computing the score for each noun with respect to the total number of their nearest adjectives in all reviews. Nouns with scores less than a particular threshold were removed and the rest were determined to be domain features. The work done by to Ghorashi et al. [12] was similar to that of Hu and Liu [10] except that they applied the H-Mine algorithm instead of the Apriori algorithm. Yang et al. [13] extracted domain features utilising a semi-automatic constructed knowledgebase that contains the top hundreds of frequently normalised nouns and noun phrases which were extracted from a collection of pre-processed reviews.

Association Rule Mining approaches extract domain features without performing human pre-processing tasks (e.g. preparing manually training dataset) because automatic natural language pre-processing is used to identify nouns and noun phrases to be domain features. However, the extracted domain features tend to be frequent domain features, whereas infrequent domain features are ignored, which can result in a reduced recall rate. In addition, some of the extracted nouns and noun phrases may not be domain features even if these occur more frequently in reviews, and this can affect the precision of the domain feature extraction task.

Machine Learning approaches require large trained datasets in order to perform the feature extraction with satisfactory accuracy. Zhuang et al. [14] extracted domain features by statistical analysis that is based on manually labelled reviews with the domain frequent features (key concepts and ground facts). Ma et al. [15] extracted domain features by training Latent Dirichlet Allocation algorithm on automatically labelled reviews with nouns or noun phrases, which were tagged via part of speech tagger and the learned domain features were expanded with synonyms, then the obtained candidate domain features were filtered by removing non-relevant domain features. Agarwal, et al. [16] extracted domain features by training a machine learning model to identify the semantic information and relations between terms in a text, which were detected by utilising dependency parse tree and Concept Net knowledgebase. Thereafter, the irrelevant domain features were removed using Minimum Redundancy and Maximum Relevance techniques. In general, Machine Learning approaches deliver significant results for domain feature extraction task using training datasets that have been manually annotated by a human expert. However, this can be an extremely time-consuming task as the required size of the training dataset should be sufficiently large to bootstrap the learning algorithms.

More recently, a new trend of studies has utilised Semantic Knowledgebase approaches that are mainly based on the knowledge of the problem domain. These approaches commonly translate the knowledge background of a chosen domain into a Semantic Knowledgebase, and then utilise this Semantic Knowledgebase to extract domain features from the pre-processed reviews. However, their approaches are different regarding the coverage of the problem domain. Zhao and Li [17] constructed a Semantic Knowledgebase that contained only the domain's key concepts and their synonyms. Martínez et al. [18] adopted a general Semantic Knowledgebase of a chosen domain that contained the domain's key concepts and their synonyms and collected ground facts from IMDb resources. Agarwal, Mittal, et al. [19] constructed a Semantic Knowledgebase for a specific domain using concepts from the top four levels of Concept Net knowledgebase, then the Semantic Knowledgebase was expanded with synonyms from WordNet.

Semantic Knowledgebase approaches have demonstrated improved performance for domain feature extraction when the knowledge of the domain of interest is utilised to extract domain features. However, the success of these techniques largely depends on the domain knowledge coverage, and the conducted investigation into the state-of-the-art approaches showed that the domain knowledge coverage is often limited and incomprehensive.

3.2. Multi-class classification of opinions

The problem of classifying opinions using a multi-point scale (also referred to as the rating inference problem) has been an interesting research area in the recent years. Early published research focused on binary classification of the overall polarity of the opinion, i.e., whether it is positive or negative [20], [21]. The obtained results of such studies indicated that Machine Learning algorithms outperformed humans on the task of binary classification of opinions [22]. More recently, researchers have focused on classifying opinions on multi-point scale rating using Machine Learning algorithms in particular supervised learning algorithms [23]. In general, these approaches are based on training a classifier on a dataset of features that have been extracted from textual contents and the corresponding target outputs (i.e., numeric rating). Then, the built classifier is tested on a dataset of features without the target outputs. Finally, the obtained outputs are compared against the real target outputs in order to evaluate the classifier [24], [25], [26]. Various techniques have been developed to improve the accuracy of the classifier's results as well as decrease the dimensionality of dataset. Lunardi, et al. [27] have proposed an approach for multiclass classification that is based on using Nested dichotomies algorithm to perform successive stages of binary classification processes. Asghar [28] has built various multi-class classifiers based on a combination of four types of extracted features (unigrams, bigrams, trigrams and latent semantic indexing) with four types of Machine Learning algorithms which are: Naïve Bayes, Perceptron Neural Networks, logistic regression and linear Support Vector Classifier. Cosma and Acampora [29] have introduced an innovative computational intelligence framework to predict customer opinions rating, which is based on using information retrieval approaches to extract features and then using an integration of Singular Value Decomposition, dimensionality reduction, genetic algorithms and different fuzzy algorithms for opinion classification on a multi-point-scale rating. The same authors have presented their updated framework via applying fuzzy C-Means and the adaptive neuro-fuzzy inference algorithms for opinion classification on a multi-point-scale rating [30]. Shirani-Mehr [31] has introduced Low-Rank Recursive Neural Tensor Networks for multi-class sentiment analysis that resulted in significant savings in computational costs. Lu et al. [32] have proposed a novel P-LSTM model for sentiment classification. P-LSTM is based on a long short-term memory recurrent neural network which uses a three-word phrase embedding instead of a single word embedding for improved classification performance.

Until recently, Machine Learning approaches have been frequently adopted for the process of opinion classification as they deliver outstanding performance, and this is because they are trained using an effective dataset of features. However, Machine Learning approaches deliver poor performance when their training dataset features are simple such as single words, character

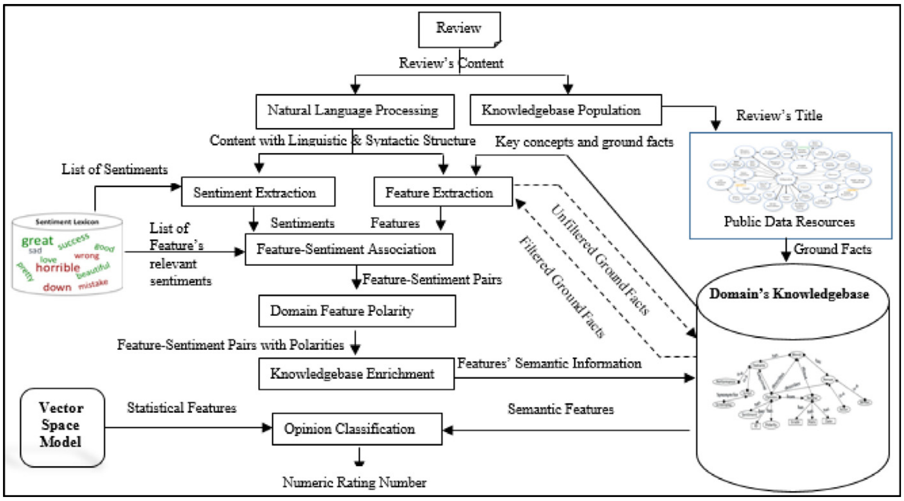
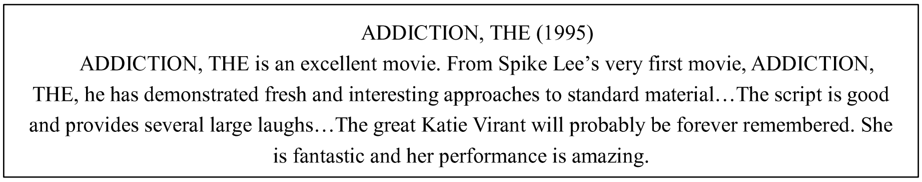


Fig. 1. A hybrid semantic knowledgebase-machine learning framework for opinion mining


ADDICTION, THE (1995)

ADDICTION, THE is an excellent movie. From Spike Lee's very first movie, ADDICTION, THE, he has demonstrated fresh and interesting approaches to standard material…The script is good and provides several large laughs…The great Katie Virant will probably be forever remembered. She is fantastic and her performance is amazing.

Fig. 2. Fictional example of movie review.

Ngrams and word Ngrams; or a combination of them [33]. The Semantic Knowledgebase approach uses a Semantic Knowledgebase that represents a shared understanding of the domain of interest, hence, the Semantic Knowledgebase approach can be used to enrich a dataset with semantic features, which can improve the performance of opinion classification task. Semantic Knowledgebase approaches rely mainly on capturing the knowledge background of a chosen domain in order to extract the domain features from reviews. These domain features are then utilised to build a Machine Learning classifier in order to classify the overall opinion of the reviews as positive or negative as in [8] and [9]. However, there appear to be no studies that investigate the use of Semantic Knowledgebase approaches to produce dataset of semantic features and then use it to build a Machine Learning classifier to classify the opinions on multi-point rating scale.

4. A new hybrid semantic knowledgebase-machine learning approach for opinion mining

The new Hybrid Semantic Knowledgebase-Machine Learning approach presents the integration of the Semantic Knowledgebase approach with Machine Learning approach to improve the performance of opinion mining process. In particular, improving the main tasks of opinion mining that include extracting domain features, associating them with their corresponding sentiments and opinion classification (i.e., solving the rating inference problem on a multi-point scale).

The new Hybrid Semantic Knowledgebase-Machine Learning approach processes unstructured textual contents, extracts domain features, and then associates the extracted domain features with relevant sentiments. Thereafter, the new Hybrid Semantic Knowledgebase-Machine Learning approach calculates the polarity for each associated feature-sentiment pair and inserts all the obtained information into a modelled domain knowledgebase. The modelled domain knowledgebase is used by the new Hybrid Semantic Knowledgebase-Machine Learning approach to further produce a semantic feature dataset, which it is merged with a statistical dataset and then used as input to Machine Learning classifier that delivers multi-point scale rating for the processed contents. Fig. 1 illustrates the architecture of the proposed new Hybrid Semantic Knowledgebase-Machine Learning approach, which comprises the following main components: knowledgebase Population, Natural Language Processing, Domain Feature Extraction, Sentiment Extraction, Domain Feature-Sentiment Association, Domain Feature Polarity, Knowledgebase Enrichment and Opinion Classification. Detailed account of the role of each component will be explained using a fictionalrunning example of a review about a movie called THE ADDICTION_1995 as is shown in Fig. 2. The following three subsections illustrate the methodology of capturing a domain knowledgebase for the proposed approach, the functionality of the proposed approach for extracting domain features and sentiments, and the procedure of associating the domain feature with their corresponding sentiments, calculating the polarity of the extracted domain feature and determining the rating class of the textual content (i.e., classification process).

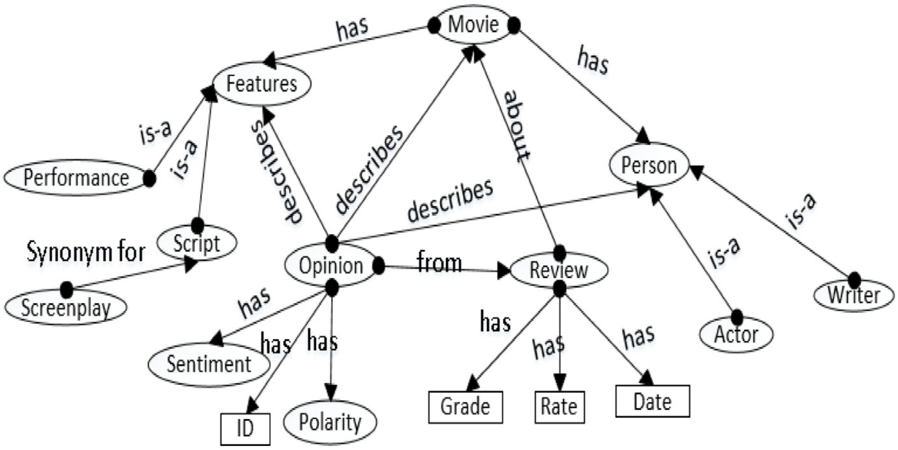


Fig. 3. Movie-opinion-review domain concept map.

4.1. Capturing domain knowledge

Opinion mining of movie reviews is considered a challenging topic because movie reviews tend to include a rich set of domain features (actors, script, plot, etc.). Furthermore, the popularity of the movie domain provides for the opportunity to exploit the ever-increasing crowd-sourced Linked Open Data repository corresponding to the movie and celebrity industry.

Using movie reviews as the target problem domain, this section describes the methodology for modelling the domain knowledge into a Semantic Knowledgebase that will be used for the domain feature extraction and opinion classification tasks. The section also describes the methodology for generating sentiment lexicons for movie's features, which will be used for associating domain features with their corresponding sentiments.

4.1.1. Semantic modelling of knowledge

Domain Knowledge is knowledge about a domain's environment that contains information such as key concepts and their synonyms and ground facts, as well as the relation between them [6]. Information from a domain's semantic knowledgebase can be utilised to improve the performance of opinion mining process, in particular, the domain feature extraction task. Constructing a semantic knowledgebase starts with modelling the domain knowledge before translating the knowledge map into formal ontologies that represent the schemata for populating the knowledgebase with structured information. The semantic structure of the knowledgebase provides for obtaining data from other public sources that use similar standards for data structuring such as Linked Open Datasets, which can be used, for instance, to populate the proposed use-case knowledgebase with dynamic ground facts about movies, actors etc.

The conceptual model for the movie-review knowledgebase was based on the key concepts of the Movie, Opinion and Review domains and the interaction (relationships) between them as illustrated in Fig. 3.

The movie-review conceptual model was translated into a formal semantic ontology that represents the template box (T-Box) of the movie-review knowledgebase. The movie-review knowledgebase comprises a comprehensive knowledge of the movie domain with approximately 504 concepts related to movie domain such as actor, sound, script, performance, etc. as well as their synonyms and the relationships between them, which were distributed as classes, object properties, data properties and annotations.

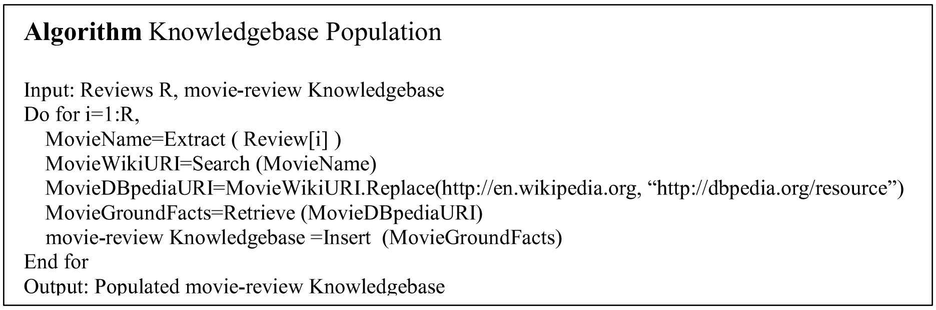
The advantage of utilising Semantic Web technologies is that they provide for the formal representation of the domain's key concepts, their synonyms and ground facts, and then link them using typed object properties (relationships). For example, the ABOUT property links a review to movie, the EXTRACTED-FROM property links an opinion to a review, and the DESCRIBE-FEATURE property links between an opinion and a movie. Moreover, there is a SYNONYM annotation for each concept and instance that has a synonym word. Such information can be used to infer valuable semantic information about the main domain concepts (such as movie) as well as the expressed opinions on its constituent features. Therefore, it is possible to compute the overall opinions about a movie across multiple reviews as well as for the cinematic features (actors, script, sound effects, etc.). For example, all movies that have a positive screenplay review can be retrieved by firing one query against the enriched movie-review knowledgebase.

4.1.2. Generating sentiment lexicons for domain features

Most opinion mining approaches involve using publically available sentiment lexicons (e.g. SentiWordNet) for the domain feature-sentiment association task. Some authors developed special sentiment lexicons for specific tasks. For example, Ghiassi et al. [34] developed a sentiment lexicon that contained sentiment terms as well as emoticons to be used for analysing Twitter messages. In the proposed Hybrid Semantic Knowledgebase-Machine Learning approach, 6800 positive and negative sentiments were obtained from a public repository opinion lexicon [35] and a sentiment lexicon was generated for each domain feature that belongs to the chosen movie reviews. Each generated sentiment lexicon contains a list of sentiments that can be used only to express

Table 1
Example of grouped movies' features and their relevant sentiments.

Key concepts and associated ground facts	Sentiments
The concept "Movie" and movies' names such as "Meet the Deedles"	Admirable, Undelivered, Horrific, Slow, Long
The concepts "Star, Writer, Editor, Director" and names of people who are stars, writer, etc.	Admirable, Able, Handsome, Gorgeous
The concepts "Writing, Screenplay, plot, script, story, idea"	Admirable, Undelivered, Well-Populated
The concept "Performance"	Admirable, Undelivered, Well, Well-Populated
The concepts "Special Effects, Visual Effects, Scene"	Admirable, Undelivered, Loud, Well-Crafted



```
Algorithm Knowledgebase Population

Input: Reviews R, movie-review Knowledgebase
Do for i=1:R,
   MovieName=Extract ( Review[i] )
   MovieWikiURI=Search (MovieName)
   MovieDBpediaURI=MovieWikiURI.Replace(http://en.wikipedia.org, "http://dbpedia.org/resource")
   MovieGroundFacts=Retrieve (MovieDBpediaURI)
   movie-review Knowledgebase =Insert  (MovieGroundFacts)
End for
Output: Populated movie-review Knowledgebase
```

Fig. 4. Knowledgebase population algorithm.

a subjective opinion for a specific domain feature. Different domain features may have a different list of sentiments. For example, the sentiment "horrific" in the sentence "It was a horrific scene" expresses a descriptive opinion on the domain feature "scene", whereas, in the sentence "It was a horrific movie" expresses a subjective opinion on the domain feature "movie". Thus, for the "scene" feature, a sentiment lexicon was generated that did not contain the sentiment "horrific", whereas it was included within a generated sentiment lexicon for the "movie" feature. Moreover, each list of sentiments can be applied to the same group of domain features. Hence, one sentiment lexicon was generated for each group of domain features that have the same classification. Table 1 shows an example of some movie's features and their relevant sentiments. Column 1 indicates a different group of movie's features, and column 2 indicates the relevant sentiments for each group.

4.2. The extraction stage

The purpose of this stage is to improve the performance of the domain feature extraction and the feature-sentiment association tasks. The extraction process exploits public data sources such as DBpedia to populate the generated movie-review knowledgebase with relevant ground facts about movies, actors, directors, prizes, etc. Then, the movie-review knowledgebase is utilised to extract the movie's features from movie reviews. The next step deploys co-referencing to identify non-explicit domain features. The movie-review knowledgebase is used also to eliminate irrelevant (i.e., false positive) domain features. Finally, the false positive opinions that express descriptive statements have been removed using the generated sentiment lexicons for each group of movie's features.

4.2.1. Knowledgebase population

The aim of populating the knowledgebase is to construct semantically structured information about each movie, which will be used for the domain feature extraction task. Thus, for each movie review, the proposed Hybrid Semantic Knowledgebase-Machine Learning approach populates the movie-review knowledgebase with the relevant ground facts (movie's name, released date, running time, country and language; movie's stars, directors, writers, editors, cinematographers, producers, etc.) that are gathered from public datasets such as DBpedia and IMDb by following the steps in Knowledgebase Population Algorithm as illustrated.in Fig. 4.

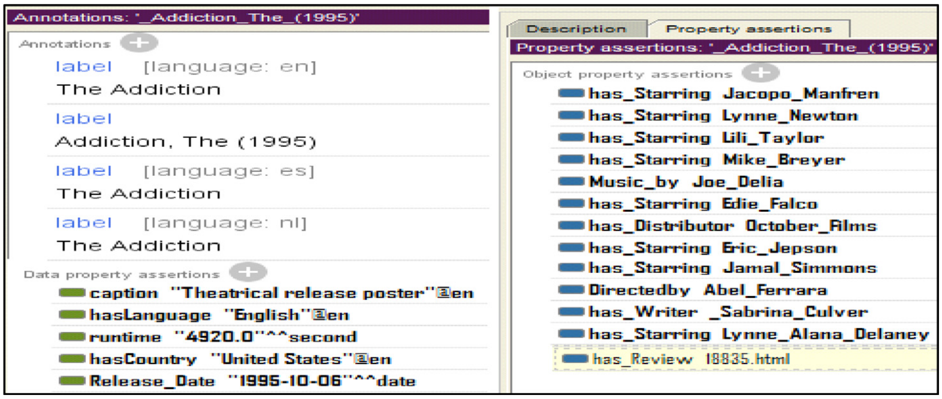
Although movie reviews are collected from the crowd-sourced data that provides extensive information with a high level of accuracy, it is likely that some movie reviews may contain incorrect information due to human error. For example, THE ADDICTION_ (1995) movie is sometimes written in the review as "ADDICTION, THE" as shown in Fig. 2. Therefore, for disambiguation, the extracted title is inserted into the movie-review knowledgebase in addition to movie's name that is retrieved from the DBpedia knowledgebase. Fig. 5 shows an example of SPARQL Construct Query for retrieving the movie's ground facts and inserting them into the movie-review knowledgebase. Fig. 6 presents a snapshot of the populated semantic information about THE ADDICTION (1995) movie into the movie-review knowledgebase.

4.2.2. Natural language pre-processing

The main objective of this process is to obtain the linguistic and syntactic structure of the textual review. Hence, Natural Language Processing tools have been implemented within the proposed Hybrid Semantic Knowledgebase-Machine Learning approach via the

```
prefix owl:<http://www.movie-review-ontology.owl#>
prefix dbpedia-owl:<http://dbpedia.org/ontology/>
prefix rdfs:<http://www.w3.org/2000/01/rdf-schema#>
prefix rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
prefix dbpprop:<http://dbpedia.org/property/>
CONSTRUCT       {     ?subject owl:movie_Title ?name .
        ?subject rdfs:label ?label .
        ?subject rdfs:label "ADDICTION,THE (1995)".
        ?subject rdf:type owl:Movie .
        ?subject owl:hasLanguage ?language .
        ?subject owl:hasCountry ?country .
        ?subject owl:has_Starring ?star .
        ?subject owl:directed_by   ?director .
        ?subject owl:edited_by      ?editor.  }
WHERE      { VALUES  ?subject {<http://dbpedia.org/resource/The_Addiction>}
     ?subject a dbpedia-owl:Film.
     OPTIONAL  {?subject rdfs:label ?label.}
     OPTIONAL  {?subject dbpprop:name ?name.}
     OPTIONAL {?subject dbpprop:language ?language.}
     OPTIONAL {?subject dbpprop:country ?country.}
     OPTIONAL {?subject dbpedia-owl:starring ?star .}
     OPTIONAL {?subject dbpedia-owl:editing   ?editor .}
     OPTIONAL {?subject dbpedia-owl:director  ?director .  } }
```

Fig. 5. Example of SPARQL construct query.


Annotations: '_Addiction_The_(1995)'
Annotations
Description
Property assertions
Property assertions: '_Addiction_The_(1995)'
Object property assertions
Data property assertions

label [language: en]
The Addiction

label
Addiction, The (1995)

label [language: es]
The Addiction

label [language: nl]
The Addiction

caption "Theatrical release poster"@en
hasLanguage "English"@en
runtime "4920.0"^^second
hasCountry "United States"@en
Release_Date "1995-10-06"^^date

has_Starring Jacopo_Manfren
has_Starring Lynne_Newton
has_Starring Lili_Taylor
has_Starring Mike_Breyer
Music_by Joe_Delia
has_Starring Edie_Falco
has_Distributor October_Films
has_Starring Eric_Jepson
has_Starring Jamal_Simmons
Directedby Abel_Ferrara
has_Writer _Sabrina_Culver
has_Starring Lynne_Alana_Delaney
has_Review 18835.html

Fig. 6. A snapshot of populated semantic information into movie-review knowledgebase about THE ADDICTION movie, which was displayed using Protégé 4.3 Framework.[1]

GATE framework [36]. Fig. 7 illustrates the linguistic and syntactic analysis including tokenisation, sentence splitting, part of speech tagging, morphological analysis, and syntax parsing, which were carried out by the Natural Language Processing components for the first sentence of the provided review in Fig. 2.

The obtained grammatical categories from these analyses are used to enhance the domain feature extraction task. For example, many words in reviews cannot be matched to the conceptualised domain features in the movie-review knowledgebase because they are found as nouns (singular and plural) or verbs. Hence, morphological analysis is performed to lemmatise each word in the review to enable the matching with the domain features via the common base. Also, as part of the Natural Language Processing process, dependency relations are analysed to determine the relation between the domain feature and a sentiment in a sentence. For example, dependency relations are used to identify adjectival and noun subject phrases respectively, which intend to contain a domain feature and its corresponding sentiment.

[1] https://protege.stanford.edu/download/protege/4.3/installanywhere/Web_Installers/.

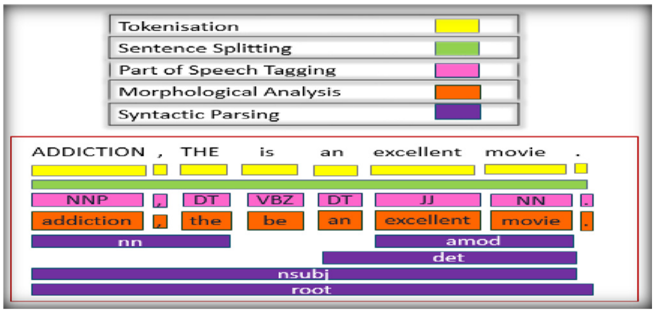


Fig. 7. Example of a processed sentence linguistically and syntactically.

4.2.3. Domain feature extraction

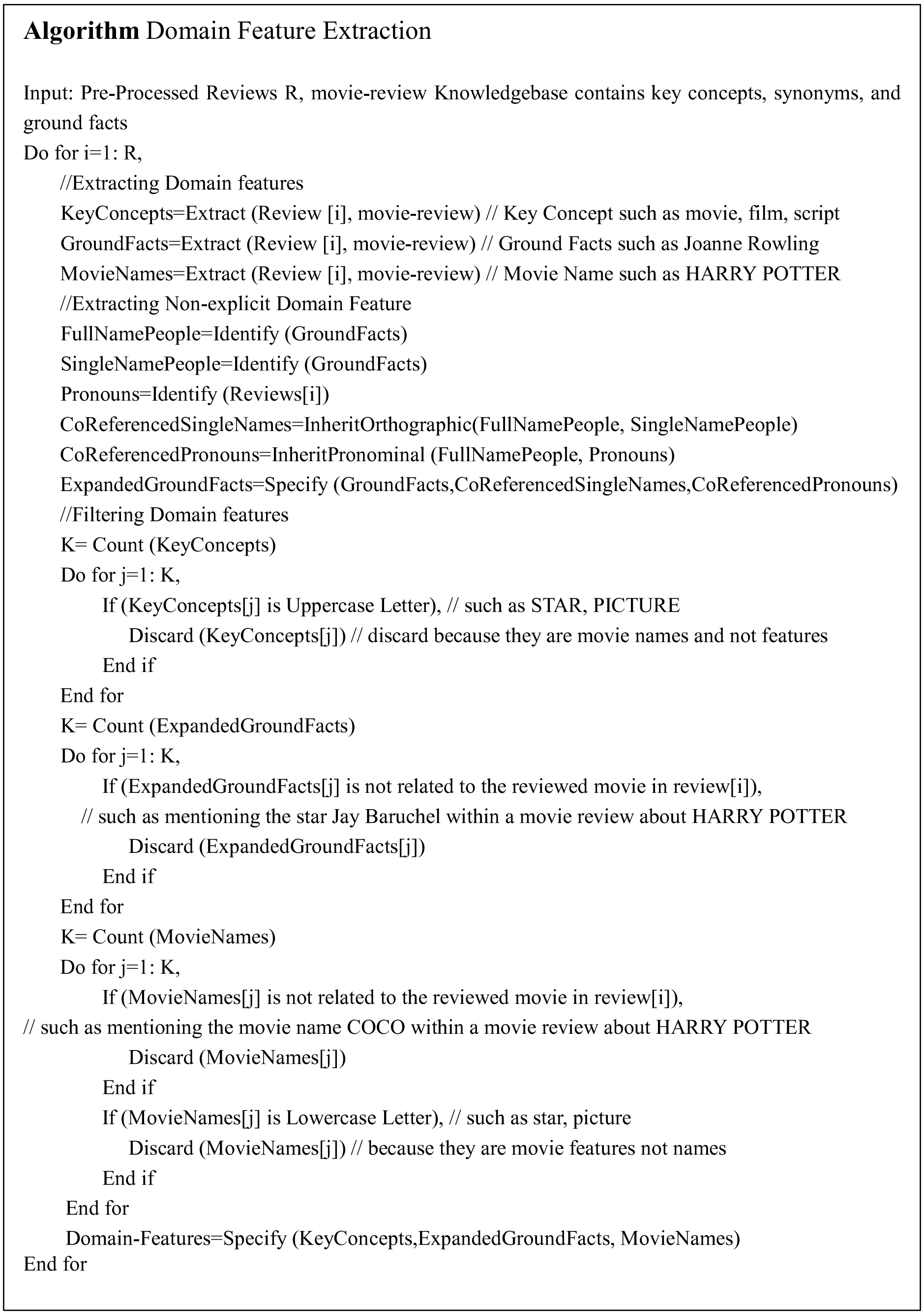
The proposed Hybrid Semantic Knowledgebase-Machine Learning approach performs the domain feature extraction task using the proposed new Domain Feature Extraction algorithm that is illustrated in Fig. 8, which is primarily driven by the movie-review knowledgebase.

As illustrated in the Domain Feature Extraction algorithm, the process contains the steps described below.

4.2.3.1. Identifying domain features by the movie-review knowledgebase. The movie-review knowledgebase was utilised to link between its conceptualised knowledge (domain's key concepts and their synonyms and ground facts) and the lemmatised words in the review. Synonyms are matched to their key concepts in the movie-review knowledgebase. For example, the word (movie) and synonymous words (film, show and picture) are matched to the same key concept (MOVIE) in the movie-review knowledgebase. Words that represent ground facts such as movie names, names of stars, writers, and editors are matched to the same individuals in the movie-review knowledgebase. In the use-case review (Fig. 2), the identified domain features by the movie-review knowledgebase are (ADDICTION THE (1995), THE ADDICTION, movie, Spike Lee, movie, ADDICTION THE, script, Katie Virant and performance) respectively.

4.2.3.2. Extracting non-explicit domain features using co-reference resolution process. Once domain features are identified by the movie-review knowledgebase, co-reference resolution is applied to identify non-explicit domain features from movie reviews such as names of people related to the movie (stars, editors, writers, etc.), which are found within the expressed opinions as single names or pronouns. According to the conducted observation in this research on movie reviews, reviewers tend to mention the full name of people at the first time of expressing opinions on them, and then only single names or pronoun are mentioned to express opinions. The conducted experiment in [37] revealed that the target domain feature is presented by a pronoun within 14% of the expressed opinions. Hence, identifying such non-explicit domain features is essential to enhance the domain feature extraction task, which leads to improving the process of opinion mining at domain feature level. The proposed co-reference resolution process is based on determining the orthographic relation between two names that refer to the same person in which one name is mentioned in a full name such as "Spike Lee", whereas the other name is mentioned in a single name such as "Lee" or "Spike". In addition, it is based on detecting the pronominal relation between a person name and a pronoun. For example, in the sentence "Spike Lee is a great director. Also, he is an amazing actor" the anaphor "he" follows the expression to which it refers, i.e., Spike Lee. Detecting the orthographic relation and pronominal relation requires a Person annotation to be generated first; this entails grouping all names (full names and single names) and pronouns (he and she) under a Person annotation, which in turn can ensure performing an accurate matching. Full names of stars, editors, writers, and so forth are matched by the movie-review knowledgebase as mentioned in the previous step, whereas single names and pronouns are identified using hand-crafted JAPE rules with the aid of GATE's named entity component called ANNIE Transducer. Secondly, GATE's co-referencing components have been used to perform matching and co-referencing between the annotated full names, single names and pronouns. Finally, the co-referenced single names and pronouns are mapped to their corresponding individuals in the movie-review knowledgebase, where the mapped individuals present full names of people who are related to movies. For example, after determining the pronominal relation, the pronouns, he and she in the mentioned review in Fig. 2 will be mapped to the director "Spike Lee" and actress "Katie Virant" respectively, which are individuals in the movie-review knowledgebase.

4.2.3.3. Filtering out the non-relevant extracted domain features using SPARQL's ASK queries. It has been observed that characteristic of reviews for movie domain is the use of uppercase letters for movie names; hence hand-crafted rules were applied to discard matched movie names that are typed in lowercase. In addition, to deal with matched movie features that are typed in upper case letters (similar to movie names). For example, in the sentence "Although Spike Lee's PICTURE, for which he won an Academy Award for the writing, is arguably his best-known film, his picture MALCOLM X, starring Denzel Washington, remains my personal favorite", the term "PICTURE" points to movie name, whereas the term "picture" is a movie's feature. Moreover, it has been observed that movie reviews contain opinions on movie's features such as (movie names and names of stars, writers, editors, etc.) that belong to the target movie as well as to other movies that are sometimes discussed in the review. Hence, the relevant semantically structured ground facts about the target movie were exploited to discard irrelevant domain features. SPARQL's ASK query was used to investigate each



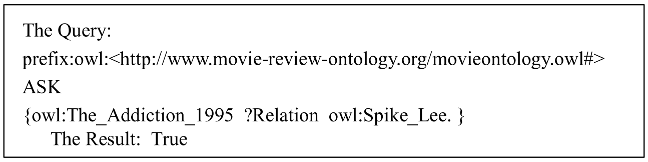
```
Algorithm Domain Feature Extraction

Input: Pre-Processed Reviews R, movie-review Knowledgebase contains key concepts, synonyms, and
ground facts
Do for i=1: R,
    //Extracting Domain features
    KeyConcepts=Extract (Review [i], movie-review) // Key Concept such as movie, film, script
    GroundFacts=Extract (Review [i], movie-review) // Ground Facts such as Joanne Rowling
    MovieNames=Extract (Review [i], movie-review) // Movie Name such as HARRY POTTER
    //Extracting Non-explicit Domain Feature
    FullNamePeople=Identify (GroundFacts)
    SingleNamePeople=Identify (GroundFacts)
    Pronouns=Identify (Reviews[i])
    CoReferencedSingleNames=InheritOrthographic(FullNamePeople, SingleNamePeople)
    CoReferencedPronouns=InheritPronominal (FullNamePeople, Pronouns)
    ExpandedGroundFacts=Specify (GroundFacts,CoReferencedSingleNames,CoReferencedPronouns)
    //Filtering Domain features
    K= Count (KeyConcepts)
    Do for j=1: K,
        If (KeyConcepts[j] is Uppercase Letter), // such as STAR, PICTURE
           Discard (KeyConcepts[j]) // discard because they are movie names and not features
        End if
    End for
    K= Count (ExpandedGroundFacts)
    Do for j=1: K,
        If (ExpandedGroundFacts[j] is not related to the reviewed movie in review[i]),
      // such as mentioning the star Jay Baruchel within a movie review about HARRY POTTER
           Discard (ExpandedGroundFacts[j])
        End if
    End for
    K= Count (MovieNames)
    Do for j=1: K,
        If (MovieNames[j] is not related to the reviewed movie in review[i]),
// such as mentioning the movie name COCO within a movie review about HARRY POTTER
           Discard (MovieNames[j])
        End if
        If (MovieNames[j] is Lowercase Letter), // such as star, picture
           Discard (MovieNames[j]) // because they are movie features not names
        End if
     End for
     Domain-Features=Specify (KeyConcepts,ExpandedGroundFacts, MovieNames)
End for
```

Fig. 8. Domain feature extraction algorithm.

matched domain feature against the relevant semantically structured ground facts in the movie-review knowledgebase as illustrated in Fig. 9. In the query, the extracted name of a person is checked whether it is relevant to the target movie or not (i.e., a star, writer, editor, director, producer or cinematographer, etc.).

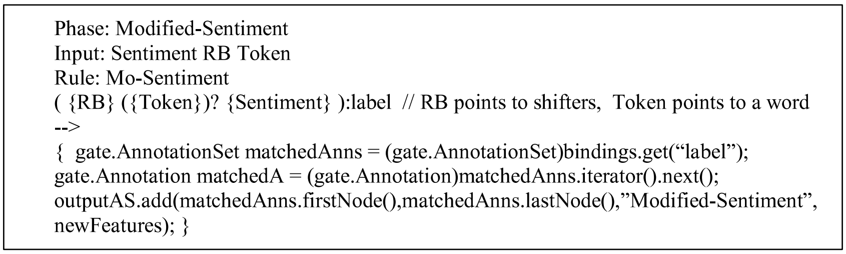
4.2.4. Sentiment extraction

In this stage, sentiments are identified using a list of sentiments (positive and negative) that were obtained from Opinion Lexicon in [35]. The particular list has been used widely in many studies [10], [38]. For example, the words (excellent, first, fresh, interesting, good, large, great, fantastic and amazing) were extracted from the provided review in Fig. 2 and annotated as sentiments. Following the identification of sentiments, any adjacent shifters (negation or adverb) were taken into account to moderate the sentiment's score



```
The Query:
prefix:owl:<http://www.movie-review-ontology.org/movieontology.owl#>
ASK
{owl:The_Addiction_1995  ?Relation  owl:Spike_Lee. }
    The Result:  True
```

Fig. 9. ASK SPARQL query for examining the relevant and irrelevant domain features.



```
Phase: Modified-Sentiment
Input: Sentiment RB Token
Rule: Mo-Sentiment
( {RB} ({Token})? {Sentiment} ):label  // RB points to shifters,  Token points to a word
-->
{  gate.AnnotationSet matchedAnns = (gate.AnnotationSet)bindings.get("label");
gate.Annotation matchedA = (gate.Annotation)matchedAnns.iterator().next();
outputAS.add(matchedAnns.firstNode(),matchedAnns.lastNode(),"Modified-Sentiment",
newFeatures); }
```

Fig. 10. Jape rules for associating sentiment with shifter.

Table 2
Dependency pattern rules.

Dependency relation	Pattern rules	Example
Nsubj : a noun phrase which is the syntactic subject of a clause	Domain feature(NN), Sentiment(JJ)	The ***movie*** is ***great***
Dobj : the noun phrase which is the (accusative) object of the verb	Sentiment(V), domain feature(NN)	I ***hate*** this ***music***
Prep-of + **NN** : Prepositional phrases followed by a noun	Sentiment(NN) + "of",Domain feature(NN)	The ***beauty*** of the ***script***
Amod : Adjectival phrase that serves to modify the meaning of the noun phrase	Sentiment(JJ), domain feature(NN)	It is a ***nice script***

NN = Noun, JJ = adjective, V = Verb.

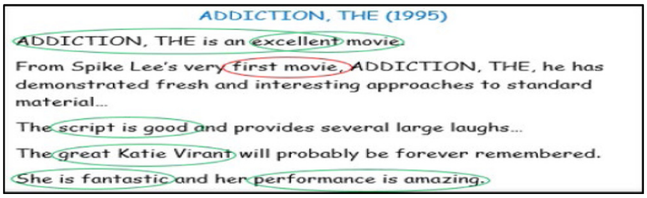
accordingly. For example, in the sentence "This is not a great movie", the shifter "not" is located nearby to the sentiment "great". Hence, the sentiment is modified to be "not great" with a score of −1. As shown in Fig. 10, the modification process was performed using hand-crafted JAPE (Java Annotation Patterns Engine) rules. JAPE provides finite state transduction over annotations based on regular expressions [39].

4.3. Classification process

The classification process implemented in this stage aims to addresses the rating inference problem on a multi-point scale (i.e., opinion classification) by integrating an enriched set of semantic data generated from the modelled movie-review knowledgebase with a statistical dataset contains the frequency of the refined term in textual reviews; the integrated data is then used to train machine learning classification algorithms.

4.3.1. Domain feature-sentiment association

In this stage, the extracted filtered domain features are associated with their corresponding extracted sentiments (feature-sentiment pairs). In other words, in each review all the mentioned statements that contain sentiments about the domain features are identified. The association process was initially performed by implementing dependency pattern rules (see Table 2) using the syntactical structure of the content where the grammatical relationships between tokens in a sentence are identified such as "amod" and "nsubj" for adjectival phrase (i.e., serves to modify the meaning of the noun phrase such as "great script") and noun subject phrase (i.e., the syntactic subject of a clause such as "the actor is good") respectively. Then, retaining only the patterns that contain both domain feature and sentiment. Finally, the associated feature-sentiment pairs (i.e., retained patterns) that hold descriptive statements were discarded using the generated sentiment lexicons for the domain features. For example, using the review example in Fig. 11, the opinion phrase "first movie" represents a descriptive statement, hence, it is discarded. Other opinion phrases such as "excellent movie, the script is good, great Katie Virant, she is fantastic, performance is amazing" represent subjective statements, and because their domain features are associated with their sentiments they are retained. A new Domain Feature-Sentiment Association algorithm, shown in Fig. 12, illustrates the association process in detail.



ADDICTION, THE (1995)

ADDICTION, THE is an excellent movie.

From Spike Lee's very first movie, ADDICTION, THE, he has demonstrated fresh and interesting approaches to standard material...

The script is good and provides several large laughs...

The great Katie Virant will probably be forever remembered.

She is fantastic and her performance is amazing.

Fig. 11. Example review illustrating the difference between descriptive statements (circled in 'red') and subjective statements (circled in 'green'). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

```
Algorithm Domain Feature-Sentiment Association
Input: Pre-processed Reviews R, Extracted domain features F, Extracted Sentiments S, Sentiment
Lexicons SL for domain features
Do for i=1: R
   Sentences=IdentifySentence (Review[i])
   DependencyPatterns=IdentifyDependencyPattern(Sentences)
  // Identify Feature-Sentiment Pairs FSPs that contains both a domain feature and a Sentiment
   FSPs=IdentifyFeature-Sentiment Pairs (DependencyPatterns, F,S)
   K=Count(DependencyPatterns)
    Do for j=1: K,
         If (DependencyPatterns[j] contains F and S)
            FSPs[j]=DependencyPatterns[j]
         Else
            Discard(DependencyPatterns[j])
         End if
    End for
// Filtering Feature-Sentiment Pairs that present descriptive opinions
      K=Count(FSPs)
      Do for j=1:K,
         If (FSPs[j] contains S that is not listed within SL for F)
            Discard (FSPs[j])
         End if
      End for
End for
Output: Filtered Features-Sentiment Pairs FSPs
```

Fig. 12. Domain feature-sentiment association algorithm.

4.3.2. Domain feature polarity

In this phase, the polarity of each extracted domain feature that has been associated with its sentiment in the previous stage is calculated using the sentiment aggregation function, which was adopted in various studies in the literature for calculating the polarity of domain features. The devised function assigns a score (weight) that indicates the proximity (distance) of the sentiment to the identified corresponding domain feature in the opinion phrase. Adopting sentiment aggregation function for domain features polarity is more effective than relying solely on syntactic dependencies that can indicate the right relation between a domain feature and a sentiment, but may not always yield accurate results, as the associated dependency patterns do not cover all the sentiments and shifters that express the opinion [40]. For example, in sentences "It is a great movie, however, it is not", "I do not think that this movie is great" and "I am not sure whether this movie is good or not", the dependency relations can be used to identify the underlined opinion phrases in order to associate domain features with their sentiments. However, the dependency relations cannot be used to accurately indicate the polarity score because they do not take into account the negation shifters.

The sentiment aggregation Score function as presented in Function (1) is based on determining the final polarity score for each extracted domain feature in a sentence.

$$\text{Score}\left(f_i, \text{s}\right) = \sum_{s_j \in s} \frac{s_j.sv}{dist(s_j, f_i)} \tag{1}$$

Let S be a sentence *(e.g.* I do not think that this movie is great*)* that contains a set of domain features f_i*(e.g.* movie*)*, and a set of sentiments s_j*(e.g.* not and great*)*; let $s_j.sv$ be the assigned score value to s_j *(e.g. not = −1 and great = +1)*; and let dist(s_j, f_i)

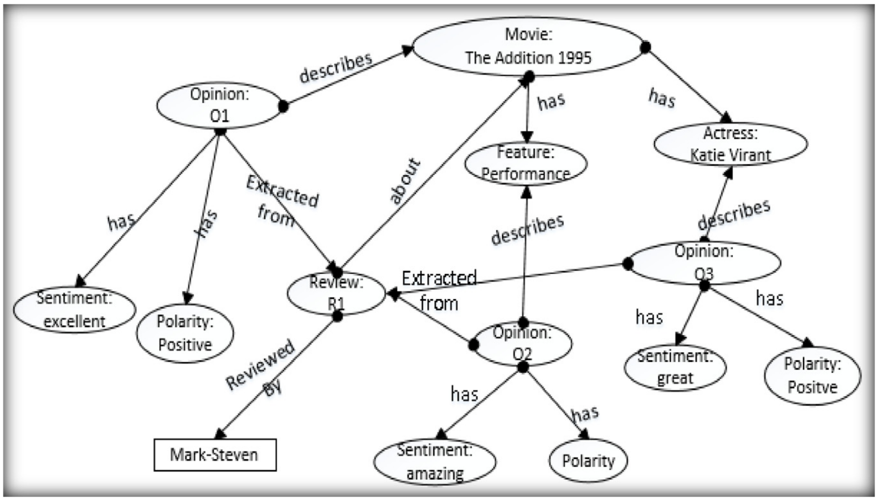


Fig. 13. A concept map for the injected semantic information into the movie-review knowledgebase.

be the distance between s_j and f_i (e.g. 4 steps between not and movie; and 2 steps between movie and great). The multiplicative inverse ($s_j.sv \div \text{dist}(s_j, f_i)$) is used to give lower weights to the extracted sentiments s_j that are far away from the extracted domain feature f_i. Finally, the domain feature f_i is assigned the final calculated polarity score after summing the multiplicative inverse of each f_i and s_j (e.g. $((-1\div4) + (1\div2)) = 0.25$) as well as it is assigned the polarity level (e.g. positive) using the condition below:

- Polarity level = Very Positive if $0.5 <$ polarity score ≤ 1
- Polarity level = Positive if $0 <$ polarity score ≤ 0.5
- Polarity level = Neutral if polarity score $= 0$
- Polarity level = Negative if $-0.5 <$polarity score < 0
- Polarity level = Very Negative if $-1 \geq$ polarity score ≤ -0.5

Dealing with negation terms or shifters such as *not*, *no*, *never*, *none*, *nobody*, *nowhere* and *neither* can be sometimes problematic when these shifters are mentioned without the associated succeeding sentiments [40]. That is because there are not any fixed rules for them. Therefore, they were treated as sentiments by assigning them a negative score value −1 and counting their distance from the specified domain feature, then aggregate them with other scores. Whereas the score of each sentiment that is preceded by a shifter in case they are adjacent such as (not good) was shifted (+1 to −1 or −1 to +1). Then, the sentiment aggregation, Function (1) was applied.

4.3.3. Knowledgebase enrichment

In this stage, the initial movie-review knowledgebase that was used to bootstrap the domain feature extraction process is further enriched with new semantic information related to the analysed review and the corresponding extracted domain features. Firstly, the review ID and the name of reviewer who wrote the review were inserted into the movie-review knowledgebase. Secondly, new semantic relations are injected into the movie-review knowledgebase for each extracted domain feature that was associated with a sentiment.

Fig. 13 illustrates a concept map for some of the injected semantic information into the movie-review knowledgebase, which is related to a review about THE ADDICTION movie. The labels in the concept map that contain “The Addition 1995”, “Katie Virant” and “Performance” indicate the movie domain’s key concepts and ground facts that were used to extract domain features, whereas, the rest of the labels indicate to the semantically-tagged information and relations about the analysed review and the extracted domain features such as the polarity level (such as very positive, positive, neutral, negative, very negative) of the extracted domain feature, and the sentiment term that was used to describe the domain feature.

The resulting movie-review knowledgebase will be accumulatively enriched with the semantically annotated movie’s features and sentiments extracted from the review, and hence will represent a valuable resource not only for predicting general opinion about a movie, but also for sophisticated retrieval of opinions associated with a specific movie’s feature. For instance, the movie-review knowledgebase should be able to answer a query about movies with the favourable screenplay, filtered by a specific genre, actor, origin, etc.

4.3.4. Opinion classification

In order to use the enriched movie-review knowledgebase for conducting the opinion classification task, it was necessary to represent the semantic information of the enriched movie-review knowledgebase in a semantic feature matrix; which will be input

```
Algorithm Opinion Classification
Input: Number of Reviews N, List of Reviews' unique identity ID, movie-review Knowledgebase
contains key concepts, synonyms, ground facts and semantic information
        //Generate Semantic Feature Matrix*/
Do for i=1: N,
    // Number of Extracted Domain features (NEDF)
    NEDF= SumDomainFeatureQuery(ID[i], movie-review)
    // Number of Positive Sentiments (NPS)
    NPS= SumPositiveSentimentsQuery (ID[i], movie-review)
    // Number of Negative Sentiments (NNS)
    NNS= SumNegativeSentimentsQuery (ID[i], movie-review)
    // Frequency of the associated Sentiments (FS)
    FS= CountSentimnetQuery(ID[i], movie-review)
    // Average Polarity for each Group of Domain features (APGDF)
    APGDF= AVGPolarityQuery(ID[i], movie-review)
    // Insert all the semantic value into a matrix
    Matrix F= Insert (NEDF, NPS, NNS, FS, APGDF)
End for
//Generate Statistical Feature Matrix
Matrix S= VectorSpaceModel(Reviews)
// Merging the matrices
Matrix FS= Merge (Matrix F, Matrix S)
// Modelling a classifier
TrainingData= Get (MatrixFS)
TestingData=Get (MatrixFS)
ClassiferModel= Build (TrainingData)
RatingResults= Test (ClassifierModel, TestingData)
Output: Numeric Rating Number for Reviews
```

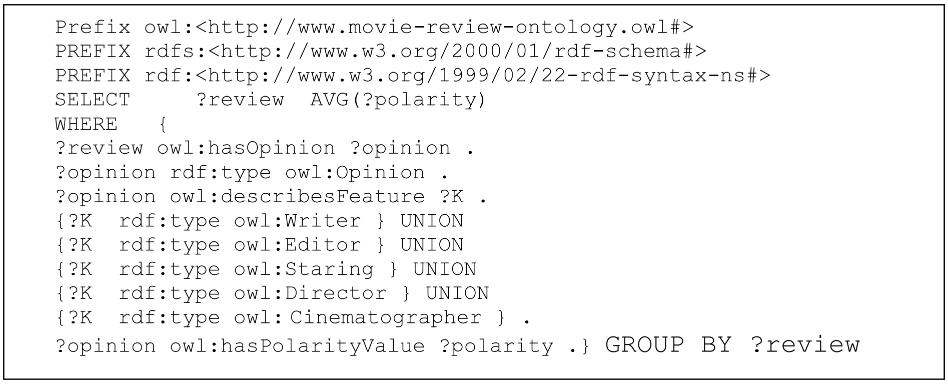
Fig. 14. Opinion classification algorithm.

into the Machine Learning algorithm after merging it with another statistical feature matrix that are generated from the same analysed reviews via applying standard Vector Space Model [41]. A new Opinion Classification algorithm illustrates the classification process in detail as showed in Fig. 14. As illustrated in the Opinion Classification algorithm, the process contains the steps described below.

4.3.4.1. Generating the semantic feature matrix. The generated semantic features present facts of the semantically structured opinions such as number of extracted domain features. Let mxn be a review by semantic feature matrix $F_{mxn} = [f_{ij}]$ where each row *i* represents a textual review, and each column *j* holds a semantic value about the extracted domain features from textual reviews. Hence, each cell f_{ij} of matrix F contains a semantic value at which a domain feature *j* appear in a review *i*. The semantic values contained in matrix F were retrieved from the enriched movie-review knowledgebase, in which each semantic value presents a specific type of information as listed below:

- Number of extracted Domain Features per a review (NDF)
- Number of Positive Sentiments mentioned in the review (NPS)
- Number of Negative Sentiments mentioned in the review (NNS)
- Frequency of each Sentiments that were Associated with Domain Features per review (FSADF)
- Average Polarity of each group of domain features (AvgP- *j*)

An example of the average polarity is that the average polarity value will be '1' for a grouped domain feature *j* in a review *i* when the grouped domain features *j* (e.g. script, story, screenplay) were extracted from a review *i* and associated with their corresponding sentiments (e.g. "the beauty of the script", "lovely story", "the screenplay was fantastic"), and their calculated polarity values are +1, +1 and +1. Although the polarity value for each extracted domain feature can be obtained via running a query on each domain feature individually, a single query was performed on each group of domain features. Grouping the domain features is based on the structure of the modelled domain key concepts in the movie-review knowledgebase. For example, the movie's features "staring, writer, editor, etc". are specified as a person, hence, instead of performing an individual query for each of them, one query is applied for these movie's features in order to combine their polarities and derive the average value as shown in Fig. 15.



```
Prefix owl:<http://www.movie-review-ontology.owl#>
PREFIX rdfs:<http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT     ?review  AVG(?polarity)
WHERE   {
?review owl:hasOpinion ?opinion .
?opinion rdf:type owl:Opinion .
?opinion owl:describesFeature ?K .
{?K  rdf:type owl:Writer } UNION
{?K  rdf:type owl:Editor } UNION
{?K  rdf:type owl:Staring } UNION
{?K  rdf:type owl:Director } UNION
{?K  rdf:type owl: Cinematographer } .
?opinion owl:hasPolarityValue ?polarity .} GROUP BY ?review
```

Fig. 15. Example of querying the average polarity of a group of domain features.

Table 3
Generated semantic features matrix.

R	F												
	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13
R1	3	2	1	1	1	0	0	0	0	0	5	0	5
R2	2	2	0	0	0	1	1	0	0	0	5	5	0
R3	2	0	2	0	0	0	0	1	1	0	1	1	0
R4	2	1	1	0	0	1	1	0	0	0	0	5	1
R5	2	2	0	0	0	1	0	0	0	1	0	5	0

The aim of grouping the polarity value of domain features is to reduce the number of zeros values in the matrix as a technique for improving the quality of matrices passed into the classifier. Prior to grouping the polarity value of domain features, the matrices were Sparse, meaning that most of their elements were zero values. Users often express their opinions on certain domain features and focus less on other domain features and this resulted in Sparse matrices. Sparse training matrices can have impact on the performance of the Machine Learning classifier because they do not contain sufficient data for training the classifier (i.e., have many zeros). Hence, minimising the zero values would improve the quality of the training data and as a consequence will improve the performance of the classifier [42]. The value of the average polarity is presented as a fuzzy value using the conditions below, where (5, 4, 3, 2, and 1 presents strongly positive, positive, neutral, negative and strongly negative respectively).

- Fuzzy Value = 5 if $0.5 <$ average polarity ≤ 1
- Fuzzy Value = 4 if $0 <$ average polarity ≤ 0.5
- Fuzzy Value = 3 if average polarity = 0
- Fuzzy Value = 2 if $-0.5 <$average polarity < 0
- Fuzzy Value = 1 if $-1 \leq$ average polarity ≤ -0.5

Table 3 presents the generated semantic feature matrix from few examples of movie reviews that are listed below. The matrix presents the total number of extracted domain features (F1), positive sentiments (F2) and negative sentiments (F3); the frequency of each sentiment that is used to express subjective opinions in each review (F4, F5, F6, F7, F8, F9 and F10 for like, beauty, great, amazing, hate, bad and nice respectively); and the average polarity of each group of domain features for example story and script are grouped together (F11, F12 and F13 for the group movie and film; the group performance and acting; and the group script and story respectively).

- Review1: I liked this movie … the beauty of the script… horrific scene.
- Review2: This movie is great … the performance is amazing.
- Review3: I hate this movie … the performance is very bad.
- Review4: The story is not great … the acting is amazing.
- Review5: This is a nice film … the acting is great.

4.3.4.2. Generating a statistical feature matrix. The generated statistical features represent the frequency of the refined terms in textual reviews. Let mxn be a review by statistical feature matrix $S_{mxn} = [s_{ij}]$ where each row *i* represents a textual review, and each column *j* holds the frequency of the refined term in textual reviews. Hence, each cell s_{ij} of S contains the frequency value (i.e., 0 for the absence or 1 for the presence) at which a term *j* appears in a review *i*. The statistical values contained in Matrix S were generated from the textual reviews as follows:

(1) Tokenising each review's contents into a list of tokens;

Table 4
Generated statistical features matrix.

R	T				
	T1	T2	T3	T4	T5
R1	1	0	0	0	0
R2	1	1	1	1	0
R3	1	0	1	0	0
R4	0	1	0	1	1
R5	0	1	0	0	1

Table 5
Merging semantic and statistical features matrices.

R	F/T									
	F1	F2	F3	...	F13	T1	T2	T3	T4	T5
R1	3	2	1	...	5	1	0	0	0	0
R2	2	2	0	...	0	1	1	1	1	0
R3	2	0	2	...	0	1	0	1	0	0
R4	2	1	1	...	1	0	1	0	1	1
R5	2	2	0	...	0	0	1	0	0	1

(2) Filtering the list of tokens by removing stop words, punctuations marks, semicolons, colons, numbers, tokens with length equal to one, tokens contain numbers and tokens that occur in only one review;
(3) Stemming the list of filtered tokens by formatting each token to its root and converting each token to lowercase letters; and
(4) Creating a Vector Space Model [41] that represents the frequency of each refined token across all reviews.

Table 4 presents the generated statistical feature matrix from few examples of movie reviews that are listed below. The matrix presents the frequency value of the refined terms that occur in more than one review (T1, T2, T3, T4 and T5 for the refined terms movie, great, performance, amazing and acting respectively).

- Review1: I liked this movie ... the beauty of the script... horrific scene.
- Review2: This movie is great ... the performance is amazing.
- Review3: I hate this movie ... the performance is very bad.
- Review4: The story is not great ... the acting is amazing.
- Review5: This is a nice film ... the acting is great.

4.3.4.3. Merging semantic and statistical feature matrices. The matrix F_{mxn} and S_{mxn} were concatenated together horizontally to produce a new matrix FS_{mxn}, in which concatenates rows of matrix F with rows of matrix S as both matrices have the same number of rows. Hence, the generated new matrix is considered a 'review by a semantic-statistical feature' matrix $FS_{mxn} = [fs_{ij}]$ where each row i represents a textual review, and each column j holds a semantic feature or a statistical feature as illustrated in Table 5. After concatenating matrices, normalisation process was performed by deploying feature scaling (i.e., each column) and instance scaling (i.e., each raw). Finally, the normalised FS matrix was passed to machine learning classifiers such as Support Vector Machine in order to result the rating inference for each review.

5. Experimental evaluation

This section presents the conducted experiments on a movie review dataset as a case study in order to evaluate the performance of the proposed Hybrid Semantic Knowledgebase-Machine Learning approach for improving the performance of domain feature extraction, the domain feature-sentiment association and opinion classification tasks. The experiments were conducted using a computer system with a processor (Intel® Core™ i5-4570 CPU @3.20 GHz 3.20 GHz), memory RAM 32.0 GB and 64-bit operating system. In addition, the GATE 8.0, Protégé 4.3, Weka 3.6 and NetBeans 8.0.1 software were installed on the system.

5.1. Datasets

A dataset of movie reviews [43] was used for the experiments, and this dataset has been widely used in the sentiment analysis literature [44], [45], [46]. The dataset contains 1770 reviews written by the same author. Each review is accompanied by a numerical rating based on its polarity. Table 6 presents the characteristics of the chosen dataset.

In order to perform evaluations for the domain feature extraction and the domain feature-sentiment association tasks, a total of 475 sentences containing 9301 words were selected from the downloaded dataset, and then from the selected sentences 422 domain features (277 Key concepts and synonyms, 18 movies' names, 91 names of people related to movies, 36 pronouns) were manually extracted and then associated with their corresponding sentiments (107 domain feature-sentiment pairs). The manually identified

Table 6
Dataset characteristics for 3-class classification.

Rating	Count
0	413
1	648
2	709
Total	1770

domain feature baseline and the feature-sentiment pairs baseline were used to evaluate the obtained domain features and domain feature-sentiment pairs via the proposed Hybrid Semantic Knowledgebase-Machine Learning approach.

In terms of the evaluation of opinion classification task, the downloaded dataset contains 1770 movie reviews and their corresponding numerical ratings for 3-class classification (0, 1 and 2). The numerical ratings of the chosen dataset will be used as reviews' rating baseline to evaluate the obtained reviews' rating via the proposed Hybrid Semantic Knowledgebase-Machine Learning approach.

5.2. Methodology

The obtained movie reviews were processed using the proposed Hybrid Semantic Knowledgebase-Machine Learning approach. Firstly, the modelled movie-review knowledgebase was populated with relevant ground facts from public datasets. Then, the movie reviews were processed linguistically and syntactically to tokenise, tag and lemmatise words as well as to determine the relation between them. After that, the target domain features were extracted from reviews and filtered to remove irrelevant extracted domain features by the proposed Domain Feature Extraction algorithm. In addition, sentiments were extracted from movie reviews and then modified to take into account any preceding shifters that might modify their polarity. Thereafter, the filtered domain features were associated with their corresponding sentiments using the proposed Domain Feature-Sentiment Association algorithm, and then their polarities were counted. Further, the movie-review knowledgebase was enriched with the obtained new semantic information and relations that belong to the processed movie reviews and the extracted domain features from them. Then, the semantic matrix F was generated from the movie-review knowledgebase, and it was merged with the statistical matrix S that was generated via standard Vector Space Model [41]. Finally, the matrix FS was normalised by deploying feature scaling (i.e., each column) and instance scaling (i.e., each raw). After that, the normalised FS matrix was used to build a Support Vector Machine and Naïve Bayes classifiers to classify each review at numerical rating scale.

5.3. Experimental results

This subsection describes the results of the domain feature extraction task, the domain feature-sentiment association task, and the opinion classification task.

5.3.1. Evaluation of the domain feature extraction task

The evaluation is based on comparing the performance of the proposed Domain Feature Extraction algorithm against the prepared domain feature baseline results as well as against two existing approaches that adopt Semantic Knowledgebase technique. In particular, three experiments were performed using the proposed Domain Feature Extraction algorithm on the same selected sentences from the downloaded reviews for three modelled knowledgebases. In the first experiment (EXP1), the modelled movie-review knowledgebase for the proposed Hybrid Semantic Knowledgebase-Machine Learning approach was utilised, which contains a comprehensive knowledge about movie domain (key concepts, synonyms and ground facts that are collected from DBpedia and IMDb resources) as described in Section 4.2.1. In the second (EXP2) and third (EXP3) experiments, two knowledgebases K1 and K2 were modelled and used as described by Zhao and Li in [17], and by Peñalver-Martinez et al. in [18]. The K1 knowledgebase contains only the movie domain's key concepts and their synonyms while the K2 knowledgebase is a general movie domain knowledgebase that contains few number of movie's key concepts, synonyms and ground facts that were collected from IMDb resources.

In all experiments (EXP1, EXP2 and EXP3), the main focus was on evaluating the number of the retrieved domain features (Recall) via different coverage of the used knowledgebase. Functions (2) was used to compute the Recall of the extracted domain features.

$$\mathbf{Recall} = \frac{|\{\text{relevant domain features}\} \cap \{\text{retrieved domain features}\}|}{|\{\text{relevant domain features}\}|} \tag{2}$$

The results indicate that the proposed Domain Feature Extraction algorithm for Hybrid Semantic Knowledgebase-Machine Learning approach achieved high overall recall (86%) even before considering the co-reference resolution in EXP1 in the case that the proposed comprehensive movie-review knowledgebase was utilised, whereas the Domain Feature Extraction algorithm achieved 64% and 57% recall in EXP2 and EXP3 when the K1 and K2 knowledgebases were utilised. In terms of the precision, the results were less relevant for critical analysis as all the experiments EXP1, EXP2 and EXP3 achieved 100% precision because all the annotated baseline domain features were extracted (i.e., relevant domain features).

EXP1 was rerun after deploying the co-reference resolution in the new proposed Domain Feature Extraction algorithm. Using Function (2), the results indicated that before applying co-referencing the recall was 86% and after co-referencing the recall increased to 93%. The number of extracted movie domain features increased by 9%, which means that non-explicit movie domain features such as single names and pronouns were matched successfully. These names and pronouns refer to people related to a movie in a particular review. Thus, the results show that deploying co-reference resolution could enhance the recall performance of domain feature extraction process especially for movie review domain, where it was observed that reviewers tend to use single names and pronouns most of the time after mentioning in the review the full name of the star, writer, editor, etc. at the first time.

Experiment EXP1 was again rerun to evaluate the impact of eliminating the non-relevant domain features by querying the movie-review knowledgebase ground facts that were obtained from public Linked Open Data sources. Functions (3) was used to compute the Precision of the extracted domain features after applying filtering process.

$$\textbf{Precision} = \frac{|\{\text{relevant domain features}\} \cap \{\text{retrieved domain features}\}|}{|\{\text{retrieved domain features}\}|} \tag{3}$$

The results evidenced that this step improved the precision of the domain feature extraction process as the number of the retrieved domain feature before filtering was 525 and after filtering was 407, and hence 118 of the retrieved were detected as non-relevant and removed. Based on the experiment EXP1, all of the 118 non-relevant domain features were movie's domain ground facts such as names of star, writer, editor, etc. As well as names of movies, however, these ground facts were determined as non-relevant because they are not relevant to the reviewed movie in a particular review. Consequently, performing a query about each extracted domain feature through the movie-review knowledgebase that was sourced via Linked Open Data resources can help to determine whether the retrieved domain feature is relevant to the reviewed movie or not.

5.3.2. Evaluation of the domain feature-sentiment association task

In this experiment, the proposed Domain Feature-Sentiment Association algorithm was evaluated against feature-sentiment pairs baseline. As described in Section 4.3.1, the proposed Domain Feature-Sentiment Association algorithm associates the extracted filtered domain features with their corresponding extracted sentiments (domain feature-sentiment pairs) using dependency pattern rules, which is similar to the approach published in [19] and [16]. The novelty of the Domain Feature-Sentiment Association algorithm is that it discards the associated domain feature-sentiment pairs that hold descriptive statements using the generated sentiment lexicons for domain features, and it retains the associated domain feature-sentiment pairs that hold subjective statements. Hence, two experiments were performed on the same selected sentences from the downloaded reviews. In the first experiment, the domain features-sentiment pairs were obtained using dependency pattern rules and without performing the filtering process, whereas in the second experiment, the domain features-sentiment pairs were obtained using the proposed Domain Feature-Sentiment Association algorithm in which the dependency pattern rules are used and the filtering process is performed. Functions (4) was used to compute the Precision of the associated domain feature-sentiment pairs within the two experiments. In function (4) DFSPs stands for Domain Feature-Sentiment Pairs.

$$\textbf{Precision} = \frac{|\{\text{relevant DFSPs}\} \cap \{\text{retrieved DFSPs}\}|}{|\{\text{retrieved DFSPs}\}|} \tag{4}$$

The domain feature-sentiment pairs associated by the proposed Domain Feature-Sentiment Association algorithm achieved the highest precision value (84%), whereas the associated domain feature-sentiment pairs using dependency pattern rules and without applying filtering process obtained a precision value of 51%. This is due to the fact that using dependency pattern rules solely could result in associating all domain features with their corresponding sentiment whether they present descriptive or subjective opinion phrases, while in the proposed Domain Feature-Sentiment Association algorithm such descriptive opinion phrases were filtered using the generated sentiment lexicons for domain features.

This research also evaluated the advantages of utilising public Linked Open Data sources on the domain feature-sentiment association task. Hence, two experiments were carried out using the same selected sentences from the downloaded movie reviews that contain the baseline associated domain feature-sentiment pairs. The first experiment (E1-KB) is based on evaluating the performance of the proposed Domain Feature-Sentiment Association algorithm when the associated domain features are the domain's key concepts and synonyms only. The second experiment (E2-KBLOD) is based on evaluating the performance of the proposed Domain Feature-Sentiment Association algorithm when the associated domain features are the domain's key concepts and synonyms in addition to the relevant ground facts that were gathered from Linked Open Data resources. Functions (5) was used to compute the Recall of the associated domain feature-sentiment pairs within the two experiments E1-KB and E2-KBLOD. In function (5), DFSPs stands for Domain Feature-Sentiment Pairs.

$$\textbf{Recall} = \frac{|\{\text{relevant DFSPs}\} \cap \{\text{retrieved DFSPs}\}|}{|\{\text{relevant DFSPs}\}|} \tag{5}$$

The obtained results evidenced that the recall of E1-KB and E2-KBLOD experiments was 69% and 73% respectively, which indicates that the number of extracted opinion phrases (associated domain feature-sentiment pairs) was increased in E2-KBLOD experiment. The improved Recall in the E2-KBLOD experiment demonstrates the benefit of populating the knowledgebase with ground facts from Linked Open Data resources which increased the number of the matched domain features and subsequently the number of the extracted opinions. Therefore, it can be concluded that populating the knowledgebase using Linked Open Data resources can enhance both domain feature extraction and feature-sentiment association processes.

Table 7
The results for 3-class classification task. (Cross validation K = 10).

Classifier	Rating class	Statistical dataset			Semantic dataset			Statistical-Semantic dataset		
		P	R	F	P	R	F	P	R	F
SVM	0	74	54	63	72	38	50	75	54	63
	1	57	65	61	51	67	58	59	69	64
	2	75	77	76	72	70	71	78	79	79
	Average	68	67	67	64	62	61	71	70	70
NB	0	67	67	67	62	47	53	70	66	68
	1	59	62	60	51	63	56	60	67	64
	2	75	71	73	72	67	70	79	73	76
	Average	67	67	67	62	61	61	70	69	70

Table 8
Accuracy of the classified reviews using the SVM and NB classifiers for the three datasets that have different number of features.

Classifier	Accuracy	Statistical dataset	Semantic dataset	Statistical-Semantic dataset
SVM	Correctly classified	67.6%	62%	70.1%
	Incorrectly classified	32.3%	37.9%	29.8%
NB	Correctly classified	67.1%	61.3%	69.7%
	Incorrectly Classified	32.8%	38.6%	30.2%

5.3.3. Results of the opinion classification task

In this experiment, the proposed Opinion Classification algorithm was evaluated for the multi-class classification of movie reviews. The aim is to determine whether adding additional semantic features to a pure dataset of statistical features can result in higher classification accuracy, as opposed to using a statistical dataset containing the frequencies of features.

The additional semantic features were generated from the movie-review knowledgebase after pre-processing the reviews by the Hybrid Semantic Knowledgebase-Machine Learning approach. For the evaluation, the reviews were classified using three (Statistical, Semantic, and Statistical-Semantic) datasets. The Statistical dataset is generated using standard Vector Space Model [41]; it contains the frequency number of each extracted word per a review (that is zero for the absence of the word or one for the present of the word). The Semantic dataset contains the valuable semantic information about the extracted domain features, which was retrieved from the enriched movie-review knowledgebase. The Statistical-Semantic dataset is a result of merging the Statistical and Semantic datasets.

Each dataset was input into the Support Vector Machine (SVM) [47] and Naïve Bayes (NB) [48] classifiers, and classification performance was evaluated. Both classifiers were tuned using the linear kernel function. The obtained results were compared against the reviews' numerical ratings on a scale [0,1 and 2] for 3-class classification. Functions (6), (7) and (8) were used to compute each of Precision, Recall and F-measure respectively for evaluating classification performance.

$$\textbf{Precision} = \frac{|\{\text{relevant reviews}\} \cap \{\text{retrieved reviews}\}|}{|\{\text{retrieved reviews}\}|} \tag{6}$$

$$\textbf{Recall} = \frac{|\{\text{relevant reviews}\} \cap \{\text{retrieved reviews}\}|}{|\{\text{relevant reviews}\}|} \tag{7}$$

$$\textbf{F} - \textbf{measure} = \frac{2\text{*Precision*Recall}}{\text{Precision} + \text{Recall}} \tag{8}$$

Table 7 presents the obtained results from two classifiers SVM and NB using the three datasets (Statistical, Semantic, and Statistical-Semantic) for 3-class classification. The results indicate that the performance of both the SVM and NB classifiers improved when they were trained using the Statistical-Semantic dataset as opposed to using the other datasets.

Table 8 presents the accuracy of the classified 1770 reviews by SVM and NB classifiers for the three datasets with respect to the number of features for each dataset, which are 1322, 716 and 2038 for the Statistical, Semantic, and Statistical-Semantic datasets respectively. Comparing the results across the various datasets when using the SVM and NB classifiers, maximum classification accuracy was consistently achieved by the SVM classifier. In particular, accuracy using SVM was 0.5%, 0.7%, and 0.4% higher for the Statistical, Semantic, and Statistical-Semantic datasets respectively, when using the SVM as opposed to when using the NB approach.

Table 9 presents the improvement of both classifiers SVM and NB when using the Statistical-Semantic dataset against Statistical dataset and Semantic dataset respectively. There was a noticeable improvement of both classifies on each of precision, recall and f-measure of the classified reviews per a rating class as well as in a total. For example, when using Statistical-Semantic dataset instead of Statistical dataset, the improvement was from +1% to +6% for the classified reviews using SVM classifier, and from +1% to +8% for the classified reviews using NB classifier. Hence, complementing the Statistical dataset with the Semantic dataset enhanced the quality of the training data and resulted in improving the performance of opinion classification task.

Table 9
Improvement of SVM and NB classifiers when using the Statistical-Semantic dataset vs. the Statistical dataset and Semantic dataset.

Classifier	Rating class	Statistical-Semantic vs. Statistical (%)			Statistical-Semantic vs. Semantic (%)		
		P	R	F	P	R	F
SVM	0	+1	+0	+0	+3	+16	+13
	1	+2	+4	+3	+8	+2	+6
	2	+3	+2	+3	+6	+9	+8
	Total	**+6**	**+6**	**+6**	**+17**	**+27**	**+27**
NB	0	+3	+0	+1	+8	+19	+15
	1	+1	+5	+4	+9	+4	+8
	2	+4	+2	+3	+7	+4	+6
	Total	**+8**	**+7**	**+8**	**+24**	**+27**	**+29**

6. Overview of the proposed approach and its applications

This paper presents a new hybrid approach for semantically extracting and analysing opinions from unstructured online reviews. The proposed approach integrates a Semantic Knowledgebase and Machine Learning approaches to improve the actionable intelligence extraction and analysis of opinions from unstructured domain reviews.

This approach comprises several stages, in which each stage was developed to improve opinion mining challenges at domain feature level. In the initial stage, we constructed a semantic knowledgebase that contains comprehensive knowledge of the problem domain. Constructing a semantic knowledgebase starts with modelling the domain knowledge into a domain model that can represent and associate generic information about the domain, opinions as well as its reviews. The domain model was then translated into a formal ontology that represents the schemata for populating the domain knowledgebase with structured information. The semantic structure of the domain knowledgebase provides for obtaining data from other public sources that use similar standards for data structuring such as Linked Open Datasets, which can be used, for instance, to populate the domain knowledgebase with dynamic ground facts about the problem domain, which is considered valuable for the process of opinion mining at domain feature level. In the second stage, we developed and implemented the domain feature extraction process to extract domain features from movie reviews. Linked Open Data resources such as DBpedia and Internet Movie Database were utilised to populate the constructed semantic domain knowledgebase with structured relevant ground facts about each processed domain review via performing composed SPARQL Construct queries. A set of Natural Language Processing components were built to obtain the linguistic and syntactic structure of the textual review. In the third stage, we developed and implemented the domain feature-sentiment association process to associate the extracted domain features with their corresponding features. Sentiment lexicon was used to extract sentiment words from the pre-processed reviews. To associate the extracted domain features with the extracted sentiments, a set of dependency pattern rules was implemented as well as a sentiment lexicon for each group of domain features was generated, which contains a list of sentiments that can be used only to express subjective opinions for a specific group of domain features. The generated domain sentiment lexicons were used to discard the identified patterns that contain descriptive opinions. In the fourth stage, the semantically structured domain knowledgebase that was used to bootstrap the domain feature extraction process was further enriched with new semantic information related to the analysed review and the corresponding semantically annotated domain features and their corresponding sentiments as well as their polarities. The resulting domain knowledgebase represents a valuable resource not only for predicting general opinion about a domain, but also for sophisticated retrieval of opinions associated with a specific domain feature. In the fifth stage, a novel hybrid Semantic Knowledgebase-Machine Learning approach was developed for classifying the overall opinion of the reviews on a multi-point scale. It is based on combining statistical features with semantic features for bootstrapping the Machine Learning opinion classifiers. The Vector Space Model was used to generate the statistical features that represent the frequency of the refined terms in textual reviews. SPARQL queries were implemented to retrieve from the developed semantic knowledgebase the semantic features that represent facts about the semantically structured opinions about domain features.

The above described methodology is applicable to other domains provided that the following two conditions is fulfilled:

(1) Construction of a semantic knowledgebase that contains comprehensive knowledge of the problem domain.
(2) The problem domain should has similar characteristics to the movie reviews domain in terms of the volume and quality of the domain's semantic information. For example, Twitter has become the most popular sources for conducting researches on sentiment analysis because it is very convenient to collect the activity of users. However, Twitter allows users to view and share limited character messages with the public, which would pose a challenge because the volume and quality of the semantic information within these posts are significantly less than within textual reviews (i.e., represent elaborate reviews written by expert critics). As in this research the analysed domain reviews represent elaborate reviews written by expert critics.

7. Conclusion

Opinions play an important role in supporting consumers make decisions about purchasing products or services. Opinion mining tools are needed to enable users to efficiently process a large number of reviews found online, in order to determine the underlying opinions. In this paper, a Hybrid Semantic Knowledgebase-Machine Learning approach is proposed where a novel Domain Feature Extraction algorithm utilises a comprehensive knowledge of the chosen domain and relevant public Linked Open Data (DBpedia, IMDb) for the purpose of improving the precision and recall of the domain feature extraction task. Our approach also improves the accuracy of computing the reviews' sentiments by deploying a new Domain Feature-Sentiment Association algorithm that relies on a generated sentiment lexicon for each domain feature. Finally, our Hybrid approach enhances the accuracy of opinion classification on a multi-point scale by deploying a new Opinion Classification algorithm that is based on integrating semantic data from a modelled semantic knowledgebase with a statistical dataset.

The experimental results for the domain feature extraction task demonstrated that the developed Domain Feature Extraction algorithm performs better with the produced movie-review knowledgebase that has more comprehensive coverage than similar reported works. Moreover, the accuracy of the proposed feature extraction algorithm was further improved by deploying co-reference resolution and by consulting the movie-review knowledgebase, which was populated utilising Linked Open Data resources, to filter out irrelevant domain features.

The experimental results for domain feature-sentiment association task evidenced that the proposed Domain Feature-Sentiment Association algorithm provided for the accurate detection of subjective opinion phrases that contain the domain features-sentiments pairs when both dependency pattern rules and the sentiment lexicons for the domain features were utilised together.

Finally, the experimental results for the opinion classification task demonstrated that the proposed Opinion Classification algorithm enhanced the classification on a multi-point scale, which answers the hypothesis of whether complementing the dataset of statistical features with knowledge-based semantic features can result in an improved classification accuracy. Future work includes an investigation on developing a SPARQL based Natural Language Query engine for sophisticated interrogation of opinions from the modelled movie-review knowledgebase. In addition, the proposed Hybrid Semantic Knowledgebase-Machine Learning approach will be evaluated on short text of movie reviews that are expressed by users as in this research the analysed movie reviews represented elaborate reviews written by expert critics.

Recent approaches on Sentiment Analysis and Opinion mining which utilise machine learning and deep learning methods focus on classification tasks. To the best of the authors' knowledge, there are no approaches which propose the adoption of a Semantic Knowledgebase approach with Deep Learning methods for classification tasks. In future work we will apply the proposed Semantic Knowledgebase approach on larger datasets to facilitate the need for Deep Learning approaches, and to evaluate the impact of the proposed Semantic Knowledgebase approach when hybridised with Deep Learning instead of conventional machine learning approaches. The proposed Semantic Knowledgebase approach can be adopted with any classifier, and the choice of classifier, whether conventional or deep learning, depends on the scale of the dataset and the task. The main contribution in this paper is based on improving the quality of the training dataset for classification tasks using a Semantic Knowledgebase approach.

References

[1] Z.H. Deng, K.H. Luo, H.L. Yu, A study of supervised term weighting scheme for sentiment analysis, Expert Syst. Appl. 41 (7) (2014) 3506–3513.

[2] M.T. Martín-Valdivia, E. Martínez-Cámara, J.M. Perea-Ortega, L.A. Ureña López, Sentiment polarity detection in Spanish reviews combining supervised and unsupervised approaches, Expert Syst. Appl. 40 (2013) 3934–3942.

[3] H.J. Min, J.C. Park, Identifying helpful reviews based on customer's mentions about experiences, Expert Syst. Appl. 39 (15) (2012) 11830–11838.

[4] M.W. Berry, S.T. Dumais, G.W. O'Brien, Using linear algebra for intelligent information retrieval, Tech. Rep. UT-CS-(1994) 94-270. [Online]. Available: citeseer.ist.psu.edu/berry95using.html.

[5] E. Kontopoulos, C. Berberidis, T. Dergiades, Bassiliades, Ontology based sentiment analysis of twitter posts, Expert Syst. Appl. 40 (2013) 4065–4074.

[6] B. Dalvi, E. Minkov, P.P. Talukdar, W.W. Cohen, Automatic gloss finding for a knowledgebase using ontological constraints, in: Proceedings of the Eighth ACM International Conference on Web Search and Data Mining ACM, 2015, p. 369.

[7] F. Ali, E. Kim, Y. Kim, Type-2 fuzzy ontology-based opinion mining and information extraction: A proposal to automate the hotel reservation system, Appl. Intell. 42 (3) (2014) 481–500.

[8] J. Polpinij, A.K. Ghose, An ontology-based sentiment classification methodology for online consumer reviews, in: Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology-Volume 01, IEEE Computer Society, 2008, pp. 518–524.

[9] A.R. Sulthana, R. Subburaj, An improvised ontology based K-means clustering approach for classification of customer reviews, Indian J. Sci. Technol. 9 (15) (2016).

[10] M. Hu, B. Liu, Mining and summarising customer reviews, in: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining ACM, 2004, p. 168.

[11] M. Eirinaki, S. Pisal, J. Singh, Feature-based opinion mining and ranking, J. Comput. System Sci. 78 (4) (2012) 1175–1184.

[12] S.H. Ghorashi, R. Ibrahim, S. Noekhah, N.S. Dastjerdi, A Frequent Pattern Mining Algorithm for Feature Extraction of Customer Reviews Extraction of Customer Reviews, 2012.

[13] C. Yang, Z. Chen, T. Wang, Sun, Research on the Sentiment Analysis of Customer Reviews Based on the Ontology of Phone, 2015.

[14] L. Zhuang, F. Jing, X. Zhu, Movie review mining and summarisation, in: Proceedings of the 15th ACM International Conference on Information and Knowledge Management ACM, 2006, p. 43.

[15] B. Ma, D. Zhang, Z. Yan, T. Kim, An LDA and synonym lexicon based approach to product feature extraction from online consumer product reviews, J. Electron. Commer. Res. 14 (4) (2013).

[16] B. Agarwal, S. Poria, N. Mittal, A. Gelbukh, A. Hussain, Concept-level sentiment analysis with dependency-based semantic parsing: A novel approach, Cogn. Comput. (2015) 1–13.

[17] L. Zhao, C. Li, Ontology Based Opinion Mining for Movie Reviews, Springer, 2009.

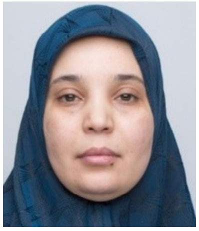
[18] I. Peñalver Martinez, F. Garcia-Sanchez, R. Valencia-Garcia, M. Rodríguez-García, V. Moreno, A. Fraga, J. Sánchez-Cervantes, Feature-based opinion mining through ontologies, Expert Syst. Appl. 41 (13) (2014) 5995–6008.

[19] B. Agarwal, N. Mittal, P. Bansal, S. Garg, Sentiment analysis using common-sense and context information, in: Computational Intelligence and Neuroscience, Vol. 2015, 2015.

[20] S.k.P. Sashi Rekha, Opinion mining from text reviews using machine learning algorithm, IJIRCCE 03 (03) (2015) 1567–1570.

[21] R. Moraes, J.F. Valiati, W.P.G. Neto, Document-level sentiment classification: An empirical comparison between svm and ann, Expert Syst. Appl. 40 (2013) 621–633.

[22] F. Sebastiani, Machine learning in automated text categorization, ACM Comput. Surv. 34 (1) (2002) 1–47.
[23] V. Vapnik, The Nature of Statistical Learning Theory, Sringer-Verlag, New York, 1995, pp. 123–171.
[24] B. Pang, L. Lee, Opinion mining and sentiment analysis, Found. Trends Inf. Retr. 2 (1–2) (2008) 1–135.
[25] R. Prabowo, M. Thelwall, Sentiment analysis: A combined approach, J. Inform. 3 (2) (2009) 143–157.
[26] H. Tang, S. Tan, X. Cheng, A survey on sentiment detection of reviews, Expert Syst. Appl. 36 (7) (2009) 10760–10773.
[27] A. Lunardi, J. Viterbo, C. Boscarioli, F. Bernardini, C. Maciel, Domain-tailored multiclass classification of user reviews based on binary splits, in: International Conference on Social Computing and Social Media, Springer International Publishing, 2016, pp. 298–309.
[28] N. Asghar, Yelp Dataset Challenge: Review Rating Prediction, 2016.
[29] G. Acampora, G. Cosma, A comparison of fuzzy approaches to e-commerce review rating prediction, in: Proceedings of International Fuzzy Systems Association (IFSA) and the 9th Conference of the European Society for Fuzzy Logic and Technology (EUSFLAT), 2015, p. 122.
[30] G. Cosma, G. Acampora, A computational intelligence approach to efficiently predicting review ratings in e-commerce, Appl. Soft Comput. ISSN (2016) 1568–4946.
[31] H. Shirani-Mehr, Applications of deep learning to sentiment analysis of movie reviews. In Technical report. Stanford University, 2014.
[32] C. Lu, H. Huang, P. Jian, D. Wang, Y.D. Guo, A P-LSTM neural network for sentiment classification, in: Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, Cham, 2017, pp. 524–533.
[33] K.P.P. Shein, Ontology based combined approach for sentiment classification, in: Proceedings of the 3rd International Conference on Communications and Information TechnologyWorld Scientific and Engineering Academy and Society (WSEAS), 2009, p. 112.
[34] M. Ghiassi, J. Skinner, D. Zimbra, Twitter brand sentiment analysis: A hybrid system using n-gram analysis and dynamic artificial neural network, Expert Syst. Appl. Int. J. 40 (16) (2013) 6266–6282.
[35] B. Liu, Opinion Mining, Sentiment Analysis, Opinion Extraction. [online] Cs.uic.edu. Available at: https://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html#lexicon [Accessed 10 Nov. 2017], 2017.
[36] Gate.ac.uk. (2017). General Architecture for Text Generation. [online] Available at: https://gate.ac.uk/ [Accessed 10 Nov. 2017].
[37] Jason Kessler, Nicolas Nicolov, Targeting sentiment expressions through supervised ranking of linguistic configurations, in: Proceedings of the Third International AAAI Conference on Weblogs and Social Media, San Jose, CA, USA, 2009.
[38] B. Liu, Sentiment Analysis, Cambridge University Press, New York, 2015.
[39] Gate.ac.uk. (2017). JAPE: Regular Expressions over Annotations. [online] Available at: https://gate.ac.uk/sale/tao/splitch8.html [Accessed 10 Nov. 2017].
[40] X. Ding, B. Liu, The utility of linguistic rules in opinion mining, in: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval ACM, 2007, p. 811.
[41] G. Salton, A. Wong, C.S. Yang, A vector space model for automatic indexing, Commun. ACM 18 (11) (1975) 613–620.
[42] J. Xu, B. Xu, P. Wang, S. Zheng, G. Tian, J. Zhao, Self-taught convolutional neural networks for short text clustering, Neural Netw. 88 (2017) 22–31.
[43] Cs.cornell.edu. (2017). Movie Review Data. [online] Available at: http://www.cs.cornell.edu/people/pabo/movie-review-data/ [Accessed 10 Nov. 2017].
[44] B. Allison, Sentiment Detection using Lexically-Based Classifiers, Text, Speech and Dialogue, Springer, 2008, p. 21.
[45] G. Li, F. Liu, Application of a clustering method on sentiment analysis, J. Inf. Sci. 38 (2) (2012) 127–139.
[46] R. Mukras, J. Carroll, A Comparison of Machine Learning Techniques Applied to Sentiment Classification, 2004.
[47] V.N. Vapnik, The support vector method, in: International Conference on Artificial Neural Networks, Springer, Berlin, Heidelberg, 1997, pp. 261–271.
[48] P. Domingos, M. Pazzani, On the optimality of the simple Bayesian classifier under zero–one loss, Mach. Learn. 29 (2) (1997) 103–130.



Rowida Alfrjani received the master's degree in Information Technology from The University of Nottingham, Nottingham, UK, in 2010, and Ph.D degree in exploiting domain Knowledge to enhance opinion mining using a hybrid semantic knowledgebase-machine learning approach form Nottingham Trent University, Nottingham, UK, in 2019. She is currently a Lecturer with Nottingham Trent University at the department of computing. Her research interests include information extraction, sentiment analysis and semantic knowledgebase field.

Dr. Taha Osman is currently a Principal Lecturer with Nottingham Trent University (NTU) and the Manager of postgraduate studies with the Department of Computing and Technology, NTU. Based on extensive research in intelligent multi-agent systems, his research expanded to knowledge-based systems, where he developed specialist expertise in utilising semantic Web technologies for intelligent retrieval and the exploration of information. His research interests include the development of innovative solutions for numerous problem domains, including financial recommender systems and Arabic sentiment analysis, and resulted in commercial R&D collaborations in the fields of digital media search engines and mineral deposits prospectivity analysis.

Dr. Georgina Cosma received the Ph.D degree in Computer Science from the University of Warwick, UK, in 2008 and a First Class Honours BSc degree in Computer Science from Coventry University, UK, in 2003. She is currently Associate Professor at Nottingham Trent University, UK. Dr Cosma is a member of the IEEE Computer Society with Computational Intelligence, Big Data Community, and Brain Community memberships. She is Principal Investigator of the Leverhulme Trust project grant entitled "Novel Approaches for Constructing Optimised Multimodal Data Spaces". Her research interests are in data science, computational intelligence, nature-inspired and deep feature selection, feature extraction, conventional machine learning and deep learning algorithms.