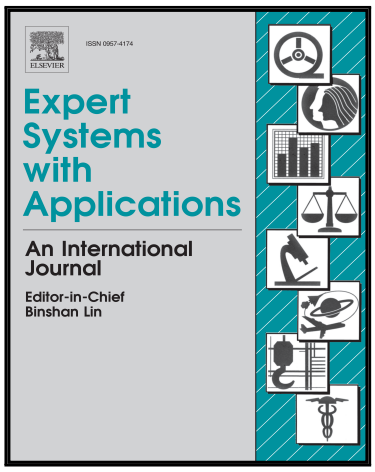

Sentiment Analysis Leveraging Emotions and Word Embeddings

Maria Giatsoglou, Manolis G. Vozalis, Konstantinos Diamantaras, Athena Vakali, George Sarigiannidis, Konstantinos Ch. Chatzisavvas

PII: S0957-4174(16)30584-X
DOI: 10.1016/j.eswa.2016.10.043
Reference: ESWA 10948

To appear in: *Expert Systems With Applications*

Received date: 4 May 2016
Revised date: 16 October 2016
Accepted date: 18 October 2016

Please cite this article as: Maria Giatsoglou, Manolis G. Vozalis, Konstantinos Diamantaras, Athena Vakali, George Sarigiannidis, Konstantinos Ch. Chatzisavvas, Sentiment Analysis Leveraging Emotions and Word Embeddings, *Expert Systems With Applications* (2016), doi: 10.1016/j.eswa.2016.10.043

Highlights

- A flexible, generic methodology for the sentiment prediction of written documents
- The methodology can be easily customized for any language
- Hybrid approach combining Word2Vec and Bag-of-Words representations
- Applied on online user reviews in both Greek and English languages
- Improved accuracy and efficiency in comparison to existing other approaches

Sentiment Analysis Leveraging Emotions and Word Embeddings

Maria Giatsoglou[a,b], Manolis G. Vozalis[b], Konstantinos Diamantaras[b], Athena Vakali[a], George Sarigiannidis[c], Konstantinos Ch. Chatzisavvas[c,*]

[a] *School of Informatics, Aristotle University of Thessaloniki, GR 54124 Thessaloniki, Greece*
[b] *Department of Informatics, Technological Education Institute of Thessaloniki, Sindos, GR 57400, Greece*
[c] *mSensis S.A., Building C2, Technopolis, GR 55535 Thessaloniki, Greece*

Abstract

Sentiment analysis and opinion mining are valuable for extraction of useful subjective information out of text documents. These tasks have become of great importance, especially for business and marketing professionals, since online posted products and services reviews impact markets and consumers shifts. This work is motivated by the fact that automating retrieval and detection of sentiments expressed for certain products and services embeds complex processes and pose research challenges, due to the textual phenomena and the language specific expression variations. This paper proposes a fast, flexible, generic methodology for sentiment detection out of textual snippets which express people's opinions in different languages. The proposed methodology adopts a machine learning approach with which textual documents are represented by vectors and are used for training a polarity classification model. Several documents' vector representation approaches have been studied, including lexicon-based, word embedding-based and hybrid vectorizations. The competence of these feature representations for the sentiment classification task is assessed through experiments on four datasets containing online user reviews in both Greek and English languages, in order to represent high and weak inflection language groups. The proposed methodology requires minimal computational resources, thus, it might have impact in real world scenarios where limited resources is the case.

Keywords: multilingual sentiment analysis, text analysis, machine learning, vector representation, hybrid vectorization, online user reviews

*Corresponding author
Email addresses: E-mail: mgiatsog@csd.auth.gr (Maria Giatsoglou), E-mail: mans@uom.gr (Manolis G. Vozalis), E-mail: kdiamant@it.teithe.gr (Konstantinos Diamantaras), E-mail: avakali@csd.auth.gr (Athena Vakali), E-mail: g.sarigiannidis@msensis.com (George Sarigiannidis), kchatz@msensis.com (Konstantinos Ch. Chatzisavvas)

Point-To-Point

Reviewer #1

---

01 *The only weakness here is that the style is not very dense: some very well known design choices could be explained in fewer words (for instance the beginning of Section 3 until 3.1, as well as 3.1.2, or 3.3, or Figure 1, which is rather elementary). Such explanations are useful to people new to the field, but not for those working already in the field.*

Following the Reviewer's suggestions we reduced the size of Section 3 until 3.1, as well as subsections 3.1.2 and 3.3.

02 *First, the results are not really compared to state-of-the-art, apart from the authors' implementation of lexical features alone or word2vec alone. To perform such a comparison, the authors could try to apply their method to the same dataset(s) used by Le and Mikolov (2014), and compare to the scores published by these authors. (It is more difficult to make sure the authors made exactly the same implementation as Le and Mikolov.) The only other method whose scores are provided is the one using Recursive Autoencoders, by Socher et al. (2011). Again, the same dataset should be used, to allow comparisons.*

It is somewhat surprising that the authors have not tried to used supervised learning to obtain lexical features automatically, such as Pang et al. (2002). Similarly, a comparison with recent systems by Mesnil et al., and Cui et al., cited in Section 2, would be important. There are also many implementations available of sentiment analysis software for English, but few are cited and none is used for comparison. Ultimately, the improvements shown are rather small for some datasets: an appreciation of the actual meaning of such an improvement should be discussed.

- Following the Reviewer's suggestions we applied our methodology in the IMDB dataset. The results has been compared with the results presented in Le and Mikolov (2014) and Mesnil et al (2015).
- We were not able to apply the RAE by Socher et al. (2011) in the IMDB dataset. The vectorization phase was extremely slow (4 weeks to complete approximately 2,000 reviews out of the 25,000 that are included in the dataset), thus, we decided to abort the respective experiment.
- We were not able to get access to the Froogle dataset that is used in Cui et al., thus, we do not present the respective comparison.
- We have updated the list of sentiment analysis software for English; we have added a new subsection 2.3 Sentiment Analysis Tools. Unfortunately none of the systems that we experimented (e.g. Stanford CoreNLP) is applicable in the case of the Greek language.

03 *Second, it is not clear whether (a) the "best" scores are the ones found to be the highest by comparing all options on the test data (after training them on the training data), or whether (b) the best parameters are first found on the training data, and the scores of the best methods are finally tested once on the test data (in which case they are not necessarily "the best"). This is a crucial point: if the authors present the best scores among all those found on test data, like in interpretation (a) above, this is not a fair comparison (it means using the test data as a second set of training data).*

Therefore, the exact way of carrying out the experiments should be clarified. For instance, the authors should provide first the best scores and parameters obtained on the training data, after learning and searching the ranges of hyper-parameters. Then, they should test this optimal solution on the test data and announce the score. Also, it is a bit puzzling that the gamma and C parameters vary so much among "best" solutions.

Following Reviewer's comments we aligned our evaluation protocol along his suggestions, as it is described in the updated 4.1.2 and it is applied in 4.2

04 *Third, it is not entirely clear what is specific to Greek sentiment analysis, and for this reason the contribution of the paper in this respect is difficult to appreciate. Apart from the richer morphology of Greek, there is little evidence that Greek behaves differently in terms of polarity-related words. If the authors think there are fundamental differences, they should make this clear, and possibly give examples of other languages, which are "more like English" vs. "more like Greek" (otherwise, every new language would deserve a new paper).*

Following Reviewer's comments we have added a paragraph in the end of the Introduction (right before the presentation of the main contributions of this work) in order to clarify the fundamental difference between weak inflection languages "more like English" and high inflection languages "more like Greek".

05 *the references in the introduction seem to miss some of the important papers in the state-of-the art, but maybe references should be left for Section 2 anyway*

We have updated the references in 2.1.1 and 2.1.2

06 *the citation style should avoid this form: "Turney (Turney, 2002) presents ..." but rather write: "Turney (2002) presents ..."*

According to ESWA's Guide for Authors
`https://www.elsevier.com/journals/expert-systems-with-applications/0957-4174/guide-for-authors#68000`,
citations in the text should follow the referencing style used by the American Psychological Association. We modified the citations form as the Reviewer suggests, where applicable.

07 *on page 16, why is only accuracy used for evaluation? Many studies use precision, recall and F-measure (averaged over the two classes), so these metrics should be used here as well.*

Following Reviewer's comments we aligned our evaluation protocol along his suggestions, as it is described in the updated 4.1.2

08 on page 17, in 4.2, there are three combinations of features (Lex1, 2, 3) but one could think of several others, so why are only three tried/reported?

Various combinations have been tested; we present Lex1/2/3 because they were the most relevant for both English and Greek languages.

09 *page 17: "(although better than the borderline of 50%)" – you probably mean "baseline" the discussion of the end of the conclusion should come earlier*

- Several typographical errors have been corrected through the manuscript
- We have moved the discussion of the end of the Conclusions in 4.2.4 Performance and Running time Comparative Results, as the Reviewer suggests.

Reviewer #2

---

10 *The authors should give the readers some concrete information to get them excited about their work. The current abstract only describes the general purposes of the article. It should also include the article's main (1) impact and (2) significance on expert and intelligent systems.*

Following Reviewer's suggestion we modify accordingly the Abstract.

11 *Please give a frank account of the strengths and weaknesses of the proposed research method. This should include theoretical comparison to other approaches in the field.*

A theoretical comparison to other approaches is given in 4.2 and summarized in 4.2.4. A brief, frank account of the strengths and weaknesses of the proposed methodology is presented in the updated Conclusions section.

12 *Moreover, I believe that it will make this paper stronger if the authors present insightful implications based on their experimental outcomes.*

We beleive that we present several insightful implications in the end of 4.2.1, 4.2.2 and 4.2.3, and also in 4.2.4

13 *You need to discuss several (say 4-5) solid and insightful future research directions*

Following Reviewer's suggestion we have added a relevant paragraph in the end of the Conclusion section.

14 *the language and grammar also require some work, and I noted a number of typographical errors.*

Several typographical errors have been corrected through the manuscript

---

List of Changes

- **Abstract**. The Abstract has been modified in order to showcase some concrete information about our work.
- **1. Introduction**.
  - We have added a paragraph in the end of the Introduction (right before the presentation of the main contributions of this work) in order to clarify the fundamental difference between weak inflection languages more like English and high inflection languages more like Greek.
  - The main contributions of this work (bullet points) have been modified in order to adopt a more logical flow of the presentation (from abstract to challenges and then to experimental). presentation flow.
- **2. Related Work**
  - Some important state-of-the-art papers have been added at the end of 2.1.2 and 2.1.3 subsubsections
  - A new subsection, i.e. 2.3 Sentiment Analysis Tools has been added in order to showcase the available sentiment analysis software for the English language.
- **3. Methodology** We reduced the size of Section 3 until 3.1, 3.1.2 and 3.3.
- **4. Experiments**
  - We modified Table 1 and the of 4.1.1 in order to present the IMDB dataset
  - We modified the whole of 4.1.2 updating the evaluation protocol according the lines suggested by Reviewer #1; also, we present precision, recall and F-score measures. Also, in 4.1.2 we add a new table, Table 2, where a comparison of SVM-Linear vs SVM-RBF models is presented, in order to choose the most efficient SVM model for the rest of our experimentation.
  - In 4.2.1, Table 2 is renamed as Table 3; we update its results using the new evaluation protocol. The respective comments have been updated, too.
  - In 4.2.2, Table 3 and Table 4 are renamed as Table 4 and Table 5. In Table 5 the results are updated using the new evaluation protocol. The respective comments have been updated, too.

  - In 4.2.3, Table 5 is renamed as Table 6; we update its results using the new evaluation protocol. The respective comments have been updated, too.
  - In 4.2.4, Table 5 is renamed as Table 7; we update its results using the new evaluation protocol. The respective comments have been updated, too. Also, we move at the end of the subsubsection the discussion of the end of the Conclusions.

- **5. Conclusions**. The Conclusions section have been modified in order to highlight the most important findings of the updated experimentations with the new evaluation protocol. Also, a brief, frank account of the strengths and weaknesses of the proposed methodology is presented. Finally, future research directions have been presented, too.

1. Introduction

On a daily basis, millions of people express their views on products, services and offers, among others, using online platforms such as social networks, blogs, wikis, discussion boards, etc. Naturally, the automatic extraction of expressed opinions or implied sentiments in the most accurate manner has become of great importance for businesses, marketing professionals and researchers.

Sentiment analysis refers to the process of identifying non-trivial, subjective information from a collection of source materials that contain latent information of people's opinions. Such a process can be applied on a variety of textual sources and on different granularity levels ranging from an entire document to individual phrases or, even, separate words. Typically, sentiment analysis reaches characterizations of positive, negative or, sometimes, neutral, for the textual sources at hand.

Sentiment analysis has become an essential part in a wide range of applications and provides benefits for multiple and diverse domains. For instance, sentiment analysis is valuable towards enhancing sales and improving a company's marketing strategies (by tracking customer reviews and survey responses), identifying ideological shifts and analyzing trends in political strategy planning, or, even, forecasting stock market momentum based on world news, financial reports and recorded social media sentiments (Bollen et al., 2011; DiGrazia et al., 2013; Ghose & Ipeirotis, 2007).

Sentiment analysis algorithms typically employ Natural Language Processing (NLP) processes (such as stemming, part-of-speech tagging, etc.) with utilization of additional resources (e.g. thesauri, sentiment- or emotion-based lexicons, sophisticated dictionaries and ontologies) to model the documents at hand. Important document features are identified towards a successful sentiment detection. Such features are, for example: the presence and frequency of terms and parts of speech, opinionated (or emotional) words and phrases, and the existence of *negations* and *intensifiers* (Medhat et al., 2014; Chatzakou & Vakali, 2015). Then, a sentiment identification step follows to characterize the textual documents based on their polarity as positive, negative, or neutral. Various techniques can be employed for the sentiment identification step which may be unsupervised or supervised. In unsupervised cases, a *lexicon-based* approach is often used; lexical resources are exploited to assign polarity scores to individual words for detecting the overall sentiment of a document. On the other hand, supervised cases typically follow a *machine learning* approach, where the sentiment detection task is considered as a classification problem by employing algorithms such as Support Vector Machines (SVM), Neural Networks or Naïve Bayes (Chatzakou & Vakali, 2015).

Recent and emerging approaches to sentiment detection take advantage of word representations embedded on semantic vector spaces (Mikolov et al., 2013; Pennington et al., 2014) that are learned through the application of neural networks (Socher et al., 2011) or probabilistic models on large text corpora (Maas et al., 2011). The derived word embeddings have been shown to accurately capture the semantics and context of words, while their use in a supervised

classification setting (especially with neural network architectures) can improve the trained sentiment models, such as e.g. in Le & Mikolov (2014); Severyn & Moschitti (2015b); Socher et al. (2011).

While lexicon-based approaches result in features that convey information on the overall sentiment orientation of a given document, they often suffer from low coverage, i.e. there are several documents that contain none of the lexicon's words. This is especially evident on short textual snippets, such as those used in online users' communication. Moreover, such approaches fail to capture more latent manifestations of sentiment and emotion, since they do not consider the context in which people express themselves. On the other hand, *word embedding-based* approaches, often employed for constructing vector representations of documents, successfully capture syntactic and semantic regularities encountered in the written language, and there are early results of their beneficial impact on sentiment classification models. However, they do not take advantage of the individual sentiment/emotion value of the words included in a document.

In this work, we present a methodology for predicting the sentiment of documents, under the hypothesis that leveraging the strength of lexicons *together* with state-of-the-art word embedding models will result in improved classification performance. Therefore, the proposed methodology derives features at the document-level using: (i) a lexicon-based and (ii) a word embedding-based approach, combined into *hybrid* vectors for a more succinct document representation. The proposed methodology is validated by a series of experiments conducted on four datasets of online user reviews (on movies and technology products, in Greek and English languages). Experiments evaluate the effectiveness of the proposed hybrid vectors in terms of sentiment detection over using separately the lexicon-based or word embedding-based feature vectors.

We aim to present a methodology that might be useful for multilingual sentiment analysis, since, in principle, the single language approach restricts the potential and the possible industrial applications of the methodology. The languages under inspection in the current research, English and Greek, have a fundamental difference as far as inflection and morphology are concerned. Modern English, is a typical weak inflection language (e.g. Swedish, Danish), while Greek is a typical high inflection language (e.g. German, Spanish) The distinction between English and Greek, is highlighted in the following short paradigms. In English there is only one form of the adjective *good* while the respective adjective in Greek $\kappa\alpha\lambda\acute{o}\varsigma$ has 11 different forms (the aforementioned adjective is also related with the sentiment analysis through the sentiment lexicons). In English, there are only 4 forms of the regular verb *ask* (*ask*, *asks*, *asked*, *asking*), while there are 93 different forms of the respective regular verb $\rho\omega\tau\alpha\omega$. Thus, we beleive that a satisfactory performance in both languages is encouraging that the methodology could be further applied in other languages.

The main contributions of this work are as follows.

- We present an abstract framework for applying sentiment analysis on different types of textual resources and different languages;

- We propose a hybrid vectorization process that takes advantage of lexicons (with polarized and emotional words) along with state-of-the-art word embedding learning approaches, and demonstrate its effectiveness over an extended set of experiments;
- We test the proposed framework on two Greek (high inflection language) and two English (weak inflection language) datasets (bibliography in the field with respect to the Greek language is still extremely limited);
- We showcase that the proposed sentiment detection framework reaches high classification accuracy in all experimentation cases, surpassing existing approaches in literature mainly in terms of efficiency;
- The methodology may support the implementation of a fast, accurate, flexible multilingual sentiment analysis application with limited computational resources.

The remainder of this article is typically organized as follows. Section 2 discusses existing related work in sentiment analysis, Section 3 presents the proposed methodology, Section 4 outlines the experimentation results and, finally, Section 5 concludes the paper.

2. Related Work

This section overviews existing research on sentiment analysis, focusing on sentiment detection overall (Section 2.1), with particular emphasis on the Greek language (a typical example of a high inflection language) as a case which imposes specific challenges (Section 2.2).

2.1. Sentiment Detection Approaches

Sentiment analysis techniques are usually divided into those employing machine learning algorithms operating under a *supervised* setting or statistically inspired methodologies under an *unsupervised* setting. Supervised approaches aim at deriving sentiment classification models, whereas unsupervised methods infer the documents' sentiment by exploiting document's statistical properties (in terms of word presence) and/or leverage existing lexicons containing polarized or emotional words. Such lexicons can, however, also be used to derive document representation features that can be used in supervised classification approaches.

2.1.1. Supervised Machine Learning Methods

Sentiment analysis is defined as a document classification problem, aiming at separating documents that express positive and negative sentiments (rarely they also take into account the neutral class) by exploiting certain syntactic and linguistic features. Recently, a limited number of more sophisticated approaches have been proposed towards identifying the, more refined, *emotion* of

the underlying documents, rather than merely its polarity (i.e. *emotion analysis*). *Emotion analysis*, or *affective analysis* can be considered as a refined version of sentiment analysis, since it aims at a more detailed categorisation of documents based on the emotions they express (Chatzakou et al., 2013). Both versions of the problem (sentiment and emotion analysis) follow similar approaches, given the existence of a suitably annotated collection of documents. Such annotations can either be nominal (e.g. *positive*, or *anger*) or real valued (e.g. 0.5 *anger* and 0.8 *fear*). Pang et al. (2002) are among the pioneers of sentiment analysis employing machine learning techniques to determine whether a written movie review is positive or negative. They experimented with three algorithms for that purpose, i.e., Naïve Bayes, Maximum Entropy and SVM, under a variety of features and parameters, such as unigrams vs. bigrams, and feature frequency vs. feature presence. Socher et al. (2011) proposed a semi-supervised machine learning framework capable of predicting multi-dimensional distributions of the underlying emotion at a sentence-level, based on Recursive Auto-Encoders (RAE). Zhang et al. (2011) proposed a hybrid technique where an augmented lexicon-based method is first applied to Twitter data to perform entity-level sentiment analysis. Then, a binary classifier receives the results of the preceding step and is trained to assign sentiment polarities to the opinionated tweets. Cui et al. (2006) showed the effectiveness of discriminative classifiers, such as SVM, using high order n-grams as features for the binary sentiment classification task. Severyn & Moschitti (2015a) discuss the use of a deep Convolutional Neural Network for sentiment classification, based on word embeddings that are initialized with the help of a unsupervised neural language model. Ren et al. (Ren et al., 2016) utilize Latent Dirichlet Allocation to obtain the topic distribution for each sentence in the dataset and then exploit Recursive Auto-Encoders to learn topic-enhanced word embeddings. To further improve performance, they integrate their representations with traditional models like SVM and logistic regression. Ensemble approaches are proposed by Mesnil et al. (2015) and Wang et al. (2015). In more detail, the work of Mesnil et al. (2015) involves blending both generative (such as Naïve Bayes and Recurrent Neural Networks) and discriminative models for sentiment prediction. The log probabilities of these models are combined via linear interpolation to extract the final sentiment assignment, surpassing all competitive models in terms of performance. Wang et al. (2015) propose a new Random Subspace method, called POS-RS, for sentiment classification based on part-of-speach analysis, which employs both content lexicon and function lexicon subspace rates to control the diversity of base learners.

2.1.2. Unsupervised Methods

These methods follow a fundamentally different approach to sentiment analysis since they do not make use of labeled documents, but rely on documents' statistical properties (e.g. word co-occurrence), NLP processes and existing lexicons with words having an emotional or polarized orientation. Turney (2002) presents an unsupervised sentiment analysis methodology which classifies reviews as positive or negative, calculating the *semantic orientation* of phrases

by associating them with only two words, *excellent* and *poor*. Lin & He (2009) propose an approach that uses Latent Dirichlet Allocation for detecting a document's sentiment and topic simultaneously, which achieves a classification accuracy similar to that of supervised approaches. There are also unsupervised approaches that rely solely on *lexicons* towards estimating the average sentiment/emotion expressed in the document by the corresponding orientation of its comprising words, according to the lexicon at hand. Such approaches often start with a small set of opinion words with known orientation, and they try to expand that set by utilizing a well known corpus or thesaurus for synonyms. They may also take advantage of structural elements and syntactic patterns that exist in the text by applying NLP processes such as lemmatization and Part-of-Speech (POS) tagging. Heerschop et al. (2011) investigated how knowledge that can be extracted from structural aspects of a document can be utilized to improve the performance of sentiment analysis. They test their hypothesis by identifying the most useful document segments for sentiment detection and score documents based on the aggregation of word-level sentiments. Qiu et al. (2010) extract opinion sentences associated with negative sentiment and find sentence topics, using a rule-based approach that combines syntactic parsing and a sentiment lexicon. They test their approach in contextual advertising, i.e. the problem of associating advertisements with a Web page. Saif et al. (2016) introduce a lexicon-based approach, called SentiCircles, that is able to update the sentiment orientation of words, by capturing their latent semantics from their co-occurrence patterns. Other works leverage NLP to identify linguistic features such as negation, intensifiers and modalities, and lexicons to identify the overall sentiment (Carrillo-de Albornoz & Plaza, 2013) or emotion of a document (Chatzakou et al., 2013), under a score averaging approach.

2.2. Sentiment Detection Application on Documents written in Greek Language

While sentiment analysis of documents written in English has become a very active research area in recent years (as indicated by the variety of approaches discussed above), there has been very little published work on sentiment analysis applications on documents written in the Greek language. However, written Greek is a particularly challenging language for NLP in general, and specifically for sentiment analysis, due to its complex morphological features (high inflection, stressing rules, etc.). A relevant effort was presented by Agathangelou et al. (2014) where an unsupervised iterative approach was employed for mining domain-specific dictionaries from sets of opinionated documents, starting with a small seed of generic opinion words. Their approach is evaluated on a set of user reviews on different types of electronic and electrical devices from a Greek e-shop. Kermanidis and Maragoudakis (Kermanidis & Maragoudakis, 2013) present an unsupervised approach for identifying the sentiment expressed in Twitter posts in order to study the degree of alignment between actual and social web-based political sentiment. They assign simple incidence and frequency values in the posts and build distinct vocabularies for different sets of posts. Solakidis et al. (Solakidis et al., 2014) take into account emoticons and lists of emotionally intense keywords in a semi-supervised emotion detection system

that is evaluated on a popular Greek student forum. Their system is tested with a number of classification algorithms that include Naïve Bayes, Logistic Regression and SVM, among others.

2.3. Sentiment Analysis Tools

There is also a growing number of tools, libraries, APIs that can be utilized for Sentiment Analysis. *Stanford CoreNLP* (Manning et al., 2014) is an integrated framework for performing NLP tasks. It includes a sentiment analysis tool that uses deep learning techniques and is trained on the *Stanford Sentiment Treebank*,[1] which includes 215,154 phrases, extracted from 11,855 sentences. *Natural Language Toolkit*[2] (NLTK) is a platfrom for building Python programs that utilize human language data. Its Sentiment Analysis tool, based on text classification, can tag a sentence as being positive, negative or neutral. To achieve that it uses classifiers trained on both twitter sentiment and movie reviews taken from the Movie Review Data.[3] *TextBlob*[4] is a Python library for processing input data in the form of texts. It includes an API for common NLP tasks, such as tagging, classification and sentiment analysis. Its Sentiment Analysis feature can return the polarity and subjectivity score for any given sentence. Other Sentiment Analysis tools are the *Sentiment API*,[5] *Sentiment140*[6] and *sentimental*.[7]

In the following section we present in detail our methodology, the vectorization techniques applied and lexicons used.

3. Methodology

The methodology of the proposed sentiment analysis framework assumes the existence of an annotated collection of documents, which belong to the same domain and are typically polarized (*positive* or *negative*) based on the respective opinions expressed. The proposed methodology is generic since it is not tied to a specific lexicon. It can rather be applied on documents written in various languages, as long as a lexicon that contains polarized or emotion words in this language is available. For a given set of documents, the desired vector representation is extracted, and a classifier is trained to derive a sentiment prediction model. Then, this model can be used to predict the sentiment of new documents of unknown polarity. The proposed methodology is highly customizable since it can function with varying types of vector representations and classification algorithms.

[1] http://nlp.stanford.edu/sentiment/treebank.html
[2] http://www.nltk.org
[3] http://www.cs.cornell.edu/people/pabo/movie-review-data
[4] https://textblob.readthedocs.io/en/dev/
[5] http://sentiment.vivekn.com
[6] http://www.sentiment140.com
[7] https://github.com/7compass/sentimental

As depicted in Figure 1 the proposed framework supports two main functionalities:

- *Model Building* This phase assumes the existence of an annotated collection of documents which will be used for training the sentiment detection model.
- *Sentiment Prediction* This phase assumes the existence of a sentiment model that has been derived via the *model building* functionality. Given one or more documents, the problem is to predict the conveyed sentiment (per document).

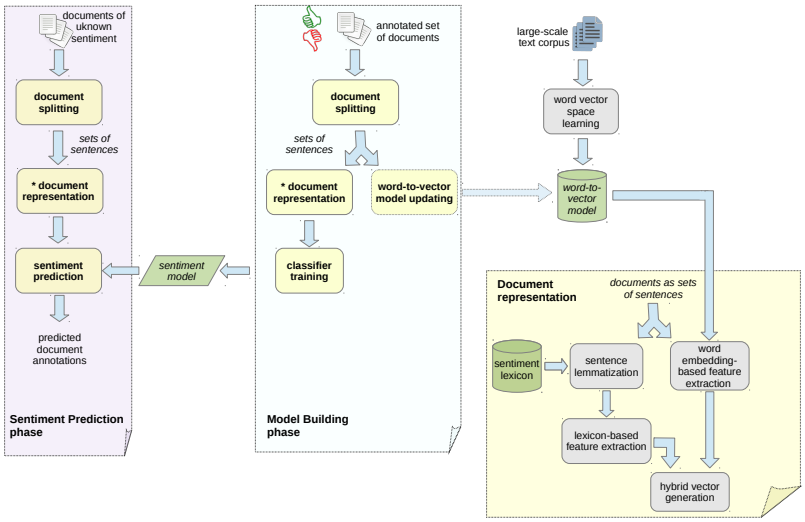

Figure 1: Sentiment Analysis Framework: Model Building and Sentiment Prediction. **Document representation* process is presented in detail in the bottom-right part of the figure.

The main advantage of the proposed framework is the extraction of the hybrid feature vectors which, as described above, is a major step needed for both the *model building* and *sentiment prediction* functionalities. Therefore, in the following subsections we will provide all the specifics about our vectorization approach. Specifically, we will give the details on the lexicon-based and the word embedding-based feature extraction methods which we utilize, as well as on the proposed hybrid vectors. Moreover, we will provide information about suitable document corpora and sentiment lexicons which are well suited for our addressed problem, focusing specifically on our *expanded Greek Sentiment Lexicon*.

3.1. Lexicon-based Features

The lexicon-based feature extraction method is based on the existence of a sentiment lexicon[8]. Typically, a sentiment lexicon consists of a set of terms in a specific language, carrying some kind of emotion weight, annotated along a number of dimensions. The number of dimensions (emotions) is lexicon-dependent (examples are presented in Section 3.1.1) while, for each dimension, a given term can be scored either in a binary manner (e.g. the term is characterized by the anger emotion or not), or by using a specific rating scale.

Terms can also be annotated regarding their *subjectivity* –i.e. classification of a document as either *subjective* or *objective*, and/or their *polarity* –which tries to answer the question whether a document is *positive*, *negative* or *neutral*. The annotation can be achieved either manually (human experts), or automatically (machine learning application on sentiment-tagged documents).

This work is inspired by the hypothesis that the richer word characterization derived using a multidimensional emotion spectrum, compared to a simple dual polarity-based scale, can improve the classification accuracy in sentiment detection; this hypothesis has also been experimentally demonstrated (Carrillo-de Albornoz & Plaza, 2013). Therefore, although our framework supports both strictly sentiment lexicons and lexicons that also include emotion dimensions, we propose the use of the latter for the sentiment detection task, when available, such as the ones described in Section 3.1.1.

3.1.1. Used Lexicons

There are a number of different sentiment and emotion lexical resources available in the English language, such as DepecheMood (Staiano & Guerini, 2014), the Subjectivity Lexicon and the Opinion Lexicon (Wilson et al., 2005), and SentiWordnet (Esuli & Sebastiani, 2006). For our purposes, we chose to use the *NRC Word-Emotion Association Lexicon*, also called EmoLex (Mohammad & Turney, 2013), due to its large size and richness in terms of the emotional dimensions used. More specifically, EmoLex includes 14,182 words tagged via crowd-sourcing in a binary manner (0=not present; 1=present) with respect to sentiment polarity (positive, negative) and the spectrum of eight emotions proposed in (Plutchik, 1994) i.e., anger, anticipation, disgust, fear, joy, sadness, surprise, and trust. Rather than comprising a list of words in their canonical form, in Emolex a given word can be found in several forms, such as "abandon", "abandoned", "abandonment", and different sentiment annotations are provided for each of them. A closer inspection of EmoLex revealed that it contains a high percentage of words with zero values in all emotion and sentiment dimensions. Such terms are not useful for our sentiment detection approach and might disorient the classification algorithm, therefore they were removed. This elimination process left a total of 6,468 words in the lexicon.

[8]The term *sentiment lexicon* refers to lexical resources that may contain emotional dimensions as well; in general, for simplicity's sake, we will use the terms *sentiment* and *emotion* interchangeably from now on.

Despite the abundance of English sentiment lexicons, other languages suffer from lack of such resources, as in the case of the Greek language. For that reason, we provide here a more detailed discussion about the Greek Sentiment Lexicon that we have utilized.

A sentiment (emotion) lexicon for the Greek language was designed by the Information Technologies Institute of CERTH (Tsakalidis et al., 2014). It contains 2,315 selected Greek terms, which have been annotated by four researchers along the dimensions of subjectivity, polarity and the six emotions proposed by Ekman (happiness, sadness, anger, fear, disgust, surprise) (Ekman, 1992). Initial experiments that we conducted based on this lexicon and the Greek *MOBILE-SEN* dataset (presented in Section 4.1.1) indicated low coverage of ∼59% (percentage of documents containing at least one of the lexicon's words), and hence the necessity for its expansion. Next, we describe the lexicon expansion process we followed, which allowed us to increase the coverage to 74%.

The lexicon's expansion was achieved via the utilization of the Altervista thesaurus[9] and a synonyms lexicon[10] which allowed us to collect a list of synonyms for each term existing in the original lexicon. Then, a list of synonyms which were already part of the lexicon was assembled for each new term and the term was assigned a vector containing the average emotion over all dimensions and terms of the list. In the end, we managed to compile the *expanded Greek Sentiment Lexicon*[11] (eGreekSentLex), which includes a total of 4,658 Greek terms, all of them accompanied with their corresponding scores.

In the preliminary Greek Sentiment Lexicon, each annotator has characterized a term with: (i) a score in the range $[1, 5]$ for the six Ekman's emotions (which reflects the term's correlation with the given emotion) and (ii) the *POSITIVE*, *NEGATIVE* or *BOTH* polarity(ies), depending on the sentiment(s) the term may convey in various contexts. When annotators were uncertain about the overall sentiment/emotion value of a given term, they assigned a "N/A" value to it. To leverage the lexicon scores in our framework, we consider nine features for each term's representation, comprising:

- the average scores over all four annotations for each one of the six Ekman's emotions,
- two scores for the positive and negative polarities taken as the average of the annotations which "voted for" the corresponding polarity value, and,
- the percentage of annotators who assigned non-"NA" scores to the term; this is considered as a *confidence score* with respect to the term's sentiment value.

[9] `http://thesaurus.altervista.org`
[10] `https://goo.gl/61ZFvQ`
[11] will be available at `http://www.msensis.com/research-and-development/demon`

3.1.2. Vectorization Techniques based on a Sentiment Lexicon

Given a set of documents (either for model building or sentiment prediction) and a sentiment lexicon in the corresponding language, the next step is the vectorization of our data. After the completion of this procedure, all documents of the dataset are transformed into vectors whose size and form may vary depending on the selected lexicon and vectorization type. The derived vectors are then fed to the selected classification algorithm for accomplishing the desired task –*model building* or *sentiment prediction*.

Based on a set of n emotional/polarized terms derived from the selected lexicon, each scored for m emotions in some rating scale, the two basic vectorization schemes that we used for our experiments are the following:

1. *Bag of Words Representation* (BoW) represented by a vector of size n
2. *Average Emotion Representation* represented by a vector of size m
3. *Mixed Representation* The combination of the aforementioned representations into a single vector whose size is $n + m$.

Next we provide some additional details on our lexicon-based vectorization process.

Vector Dimensionality. In order to exclude words carrying little or no information and to limit the size of the resulting vectors, we set a *minimum appearance threshold*, t. Words from the corresponding sentiment lexicon appearing in the selected dataset less times than t, are removed from the final sentiment lexicon set. The minimum appearance threshold is clearly dataset dependent and, thus, empirically set. Therefore, in the *BoW representation* the actual dimensionality of the vectors is $l << n$, where l is the number of lexicon words that appear more times than t in the dataset used for *model building*. These are also the (lexicon-based) features that are used for the representation of the documents in the *sentiment prediction* phase.

Negation Handling. Proper treatment of *negation* is important for sentiment analysis methods (Lapponi et al., 2012; Wiegand et al., 2010), and has been shown to improve sentiment classification results (Taboada et al., 2011; Carrillo-de Albornoz & Plaza, 2013) (the level of improvement depends on the frequency with which negation appears in the dataset at hand). In this work, to address negation, we initially compiled a list of negation words for the English (e.g. no, never, don't) and Greek language (e.g. $\mu\eta\nu$, $\delta\epsilon\nu$, $\acute{o}\chi\iota$). Then, each time a word from the sentiment lexicon set was located in a document from the dataset, we checked in a predefined range of preceding words called *negation window* (empirically set at 2) for a word belonging to the negation list. If such a word was found, we followed one of the next two approaches.

1. *REVERSE approach*
   Reverse the feature values to their negative counterpart –being either equal to 1 under the binary scoring manner, or a real-valued emotion rating, otherwise.

2. *DOUBLE approach*
Double the size n of the sentiment lexicon set and the size m of the lexicon's emotions, thus also the dimensionality of the constructed vectors. The additional n dimensions are derived from the treatment of the negated lexicon terms encountered in the documents as distinct features, while their m emotional dimensions are also considered as different, negated emotions.

Example
Suppose a lexicon that uses the Ekman's emotion spectrum and characterizes the word *happy* with the scores 0.8, 0.05, 0.1, 0.04, 0.1, 0.7 for happiness, sadness, anger, fear, disgust and surprise, respectively. Let this word be encountered in a document in its negated form (e.g. *not happy*). Following the *REVERSE* approach, we will assign the value -1 to the feature corresponding to the word "happy" for the *BoW representation*. For the *Average Emotion representation* we will consider the scores -0.8, -0.05, -0.1, -0.04, -0.1, -0.7 in the calculation of the average features for happiness, sadness, anger, fear, disgust and surprise. Following the *DOUBLE* approach, we will assign the value 1 to the happy_negated feature for the *BoW representation*, and include the scores 0.8, 0.05, 0.1, 0.04, 0.1, 0.7 when calculating the average features for happiness_negated, sadness_negated, anger_negated, fear_negated, disgust_negated and surprise_negated for the *Average Emotion representation*.

It is worth mentioning that since the *DOUBLE approach* typically results in very sparse vectors, uninformative features (i.e. having zero values in all documents) are removed. Thus, at the end, the vector dimensionality is $n' + 2 \times m$, where $n < n' < 2 \times n$.

Greek Language Handling. As mentioned before, the Greek language has special characteristics according to which the same word may appear in a large number of forms in text, while to further perplex the situation, text created freely by online users is often of poor quality including spelling errors and missing intonation. In order to facilitate the matching between the terms contained in the lexicon and those contained in the input documents, and thus assist the sentiment classification task, we apply a text lemmatization process: we identify the canonical form or *lemma* for each correctly typed Greek word in the dataset by removing all morphological alterations due to inflection. For this task we employed our in-house Greek dictionary[12] which includes about 60,000 lemmata along with all inflection forms of all the nouns, verbs and adjectives in the dictionary. Moreover, the dictionary keeps two forms for each word, with and without intonation, so we can find the best match for those words missing intonation. Various tools have been utilized in order to succeed the best possible completeness of our dictionary (Kotrotsios, 2015).

[12]will be available at `http://www.msensis.com/research-and-development/demon`

3.2. Word Embedding-based Features

Although lexicon-based features can be used to provide an overall indication of a document's sentiment, they can not capture more refined characteristics and contextual cues that are inherent in the human language. People often express their emotions and opinions in subtle ways (such as e.g. when they use irony), mix positive with negative polarities or diverse emotions in their expressions, or rely on a set of context-specific expressions/word to communicate their opinion. Recently proposed word embedding-based approaches try to capture semantic and syntactic features of words out of document collections in a language independent process.

3.2.1. Word2Vec Method

Word2Vec (Mikolov et al., 2013) is an (unsupervised) word embedding-based approach. It aims to detect the meaning and semantic relations between words by exploiting the co-occurrence of words in documents belonging to a given corpus. The core idea of Word2Vec is to capture the *context* of words, using machine learning approaches such as Recurrent or Deep Neural Networks. It actually involves two different learning algorithms: (i) the *Continuous Bag-of-Words* algorithm (CBOW) –whose goal is to predict a word when the surrounding words are given, and (ii) the *Continuous Skip-Gram* algorithm (Skip-gram) –which predicts a set of words when a single word is known. According to Mikolov et al. (2013), *Skip-gram* operates well with a small amount of training data, representing accurately even rare words or phrases, whereas *CBOW* is much faster to train (than *Skip-gram*) and slightly more accurate for frequent words.

Word2Vec operates on a corpus of sentences, first constructing a vocabulary based on the words that appear in the corpus more times than a user-defined threshold (to eliminate noise), and then applying either the *CBOW* or the *Skip-gram* algorithm on the input documents to learn the words' vector representations in a D-dimensional space[13]. Large textual corpora (e.g. all available Wikipedia articles in a certain language) are often used for training Word2Vec, capturing strong linguistic regularities in the words' relative positions in the learned word vector space that are of a global scope (within the given language). Alternatively, or additionally, thematic textual collections can be used to train Word2Vec, so as to better imprint the use of words in a given domain.

3.2.2. Vectorization Approach based on Word2Vec

Our framework leverages Word2Vec to generate features that best capture the context of the documents whose polarity we need to predict. After the text preprocessing step, each document d comprises k sentences $s_{d,i}$, $i \in [1, k]$. In the *model building* phase, before training the classifier, we can either use these sentences to learn a new word vector representation via Word2Vec, or

[13] D is user-defined. Generally, it has been shown that as few as 50-300 dimensions are enough to model hundreds of millions of words with high accuracy (Mikolov et al., 2013).

use an existing Word2Vec model pre-trained on a corpus of large scale and thematic coverage. In the second case, we can optionally continue the training of the Word2Vec model on the input sentences in order to update its weights for the existing word representations, given the new, domain-specific, information. In all cases, the derived word-to-vector mapping is used to extract a vector representation for each word encountered in the input sentences. We then follow a simple approach, according to which we derive the vector representation of a given a sentence $v_{d,i}$ as the average over the vectors of all its comprising words. Similarly, we take the average of all $v_{d,i}$ vectors as the vector representation for the document d. This approach maintains a fixed and relatively small size for the documents' representation. The same process takes place in the *sentiment prediction* phase to derive the vector representations of the new documents. The only difference is that a pre-trained (domain-specific or generic) model is used and we do not continue its training on the new data. We have to mention that in the document representation phase, since Word2Vec exploits the sentences' syntax and structure, we do not apply lemmatization on them in order to capture the different uses of the varying word forms depending on the context.

3.3. Hybrid Features

In this work we propose a hybrid vectorization process that takes advantage of both lexicon-based features and word embedding learning approaches. Based on this process, a new **hybrid vector** is constructed for the representation of each document through the concatenation of the lexicon-based and word embedding-based feature vectors. Feature vectors are generated with the processes described in Section 3.1 and Section 3.2, respectively. The use of *hybrid vectors results to document representations that capture the overall emotion/sentiment orientation of the document*, which is lacking from strictly word embedding-based representations. Also, it leverages the latter's contextual and semantic expressive power.

We present experiments on various types of hybrid vectors based on the selected lexicon- and word embedding-based vectorization processes in Section 4.

4. Experiments

In this section, our proposed sentiment detection approach is evaluated through a series of experiments on four datasets of user reviews from online web sites in both Greek and English. We provide details on the datasets used, our experimentation approach, the settings of our experiments, and finally we present the performance of our methodology in terms of effectiveness and efficiency.

4.1. Experimental Setup

The evaluation of our proposed methodology requires a diverse set of use cases as far as documents' language, length, and domain are concerned. To this

end, we resorted to the use of four sets of online user reviews, which are all challenging due to the informal and non-standard expressions used by people in their online activities/communication, and they have different characteristics in terms of language (two of them are in Greek and the other two in English), in terms of length (two datasets contain annotated sentences and the other two paragraphs), and domain (the reviews are on movies, mobile phones and other electronic and electrical devices). Aiming primarily to assess the possible advantage of hybrid vectors over their lexicon- and word embedding-based counterparts, we experimented with different hybrid vector generation approaches and compared them in terms of their derived accuracy and execution time over all datasets.

4.1.1. Datasets

Due to the limited existing work on sentiment detection on the Greek written language and the challenges it presents (as discussed in Section 2), we placed special emphasis on experimenting with collections of Greek documents to guarantee the suitability of our approach for this case. To our knowledge, there is only one publicly available dataset that contains documents in the Greek language and has been used for sentiment analysis (Agathangelou et al., 2014). The dataset contains reviews from a popular, Greek e-shopping site[14] about various products (TVs, Air Conditioners, Washing Machines, Cameras, Refrigerators, Mobile Phones and Tablets), with each review having a score ranging from 1-5. Authors in Agathangelou et al. (2014) followed a typical approach for the ground-truth sentiment extraction from the reviews, according to which, all reviews which have a score in [1,2] ([4,5]) are considered to be negative (positive). The reviews in this dataset comprise one or more sentences (i.e. they are paragraphs).

In order to compare our supervised approach with the unsupervised approach presented in Agathangelou et al. (2014), where the evaluation in terms of sentiment detection is based on the accuracy derived on each individual product category, we indicatively focused on predicting the sentiment of a single category, namely Mobile Phones (which is also the biggest one in the dataset). Thus, we decided to use all reviews in the dataset for training, except for those corresponding to the Mobile Phones category that we retained for testing. This approach resulted in a heavily unbalanced training dataset (1,012 positive and 172 negative reviews), therefore we balanced the dataset by oversampling the minority class (maintaining the same participation ratio for each category in the training data). The final dataset *MOBILE-PAR* contains 1,976 reviews for training and 3,329 for testing (Table 1).

Moreover, we created a more focused, new dataset *MOBILE-SEN* by extracting reviews from the same Greek e-shopping web site on the domain of Mobile Phones gathering in total 1,768 reviews on 176 products. We split the reviews into sentences and tagged them manually for their positive or negative

[14] `https://www.skroutz.gr/`

Table 1: Datasets used for experimentation. The number of Training and Test documents corresponds to a single fold (when applicable). The percentage in the parentheses corresponds to the positive class.

Dataset	Language	Training Documents (pos%)	Test Documents (pos%)	Folds
MOVIES	English	9,596 (50.0%)	1,066 (50.0%)	10
IMDB	English	25,000 (50.0%)	25,000 (50.0%)	-
MOBILE-SEN	Greek	2,520 (50.0%)	280 (50.0%)	10
MOBILE-PAR	Greek	1,976 (51.2%)	3,329 (84.6%)	-

sentiment. In the end, we kept a total of 2,800 sentences, having a balanced class distribution, as depicted in Table 1, and generated 10 folds in order to perform cross validation during the experimentation process.

As far as the English language is concerned, we used a popular dataset (Pang & Lee, 2005) comprising sentences extracted from a movie reviews website (Rotten Tomatoes[15]). The polarity of the sentences was automatically determined, depending on whether the review in which a given sentence belongs to was marked as positive or negative by the author. The dataset *MOVIES* comprises a total of 10,662 sentences and was split into 10 folds, as depicted in Table 1, to suit the experimentation needs. Finally, we used another standard English dataset (Maas et al., 2011). The Large Movie Review Dataset[16] (*IMDB*) comprises a total of 25,000 reviews for training and 25,000 reviews for testing.

4.1.2. Evaluation Protocol

We instantiate our framework with two typical SVM classifiers: one with a linear kernel (SVM-Linear) and another one that uses a Gaussian radial basis function as a kernel (SVM-RBF). Before training the SVM models, we first standardize the datasets using Min-Max feature scaling in the $[-1, 1]$ range (as described in Section 3.3). Then, we conduct experiments on the datasets presented in Section 4.1.1 to evaluate our methodology in terms of effectiveness and efficiency.

We use a 10-fold cross validation approach splitting the dataset into training and test subsets, whenever this is possible. For example, *IMDB* and *MOBILE-PAR* datasets are already been split into training and test datasets, so no cross validation is used there. However, in all cases in order to find the best classification model we split the training set into training and validation subsets using 5-fold cross validation.

The perfomance is assessed based on *accuracy* on the test set, defined as:

$$\text{accuracy} = \frac{\#\text{true\_pos} + \#\text{true\_neg}}{\#\text{true\_pos} + \#\text{false\_pos} + \#\text{true\_neg} + \#\text{false\_neg}} \quad (1)$$

[15] `http://www.rottentomatoes.com/`
[16] `http://ai.stanford.edu/~amaas/data/sentiment/`

Table 2: Comparison between SVM-Linear vs SVM-RBF on MOBILE-SEN dataset

Vectorization/ Corpus	SVM-Linear Accuracy (%)	SVM-RBF Accuracy (%)
Lex1	72.79	74.88
Lex2	73.32	75.00
Lex3	63.39	66.43
MOB-GR	71.79	69.76
WIKI-GR	69.79	70.83
WIKI-MOB-GR	70.89	70.60
W2V+Lex1	74.36	75.71
W2V+Lex2	74.93	76.79
W2V+Lex3	71.43	73.21
W2V+Lex1	78.00	75.95
W2V+Lex2	78.57	77.62
W2V+Lex3	71.79	72.14

Also, we calculate *precision*, *recall*, and we present their harmonic mean, the balanced *F-score*:

$$\text{F-score} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{2}$$

where

$$\text{precision} = \frac{\#\text{true\_pos}}{\#\text{true\_pos} + \#\text{false\_pos}} \quad \text{and} \quad \text{recall} = \frac{\#\text{true\_pos}}{\#\text{true\_pos} + \#\text{false\_neg}}$$

For the parameterization of the SVM[17] we use grid search on the parameters C (and *gamma* for SVM-RBF).

For the evaluation of the compared approaches in terms of efficiency, we use the running time required for training and prediction in a single fold.

Aiming for the best possible performance of the proposed methodology in terms of accuracy and running time, we test both SVM models in our datasets. For all datasets the running time of the SVM-Linear is almost one order of magnitude lower compared to SVM-RBF, while there is no superior model in terms of accuracy. In Table 2 an indicative example of the comparison between the two SVM models (applied on MOBILE-SEN) is presented. For all cases presented in Table 2 the ratio $\frac{t_{\text{SVM-Linear}}}{t_{\text{SVM-RBF}}} < \frac{1}{6}$. Thus, we choose the SVM with the linear kernel for our methodology.

Finally, we also report effectiveness and efficiency results derived on the *MOVIES*, *MOBILE-SEN* and *MOBILE-PAR* datasets when a well-known algorithm based on Recursive Auto-Encoders (RAE) (Socher et al., 2011) was applied. This algorithm learns vector representations of phrases/sentences and their hierarchical structure, as well as an emotion distribution for each structure's node. For the *IMDB* dataset we compare our results with the best results

[17]We used the LibSVM implementation (Chang & Lin, 2011).

provided in the literature (Mesnil et al., 2015) and (Le & Mikolov, 2014), according to our knowledge.

4.2. Results

In the following paragraphs, we present the results derived from the application of the previously described evaluation protocol on the three datasets. First, we experiment with different lexicon-based (Section 4.2.1) and word embedding-based vectorization approaches (Section 4.2.2), separately, in order to evaluate them on the basis of the derived classification outcome. Then, we select the best of these approaches and use them for the generation and evaluation of the proposed hybrid feature vectors. The results of these experiments are presented in Section 4.2.3). Finally, we compare all considered approaches with respect to their performance and required running time, along with a well-known sentiment analysis method, in Section 4.2.4.

4.2.1. Experiments on Lexicon-based Vectors

This set of experiments aims to examine the effect of the representation scheme used for generating the lexicon-based features on the derived classification accuracy. We first describe the different approaches tested on the datasets. The lexicon-based vectors are extracted based on Emolex and eGreekSentLex, for the English and Greek datasets accordingly, with the following three approaches in terms of comprising features:

Lex1 *Mixed Representation* (comprising *BoW-REVERSE* and *Average emotion-REVERSE* representations);

Lex2 *Mixed Representation* (comprising *BoW-DOUBLE* and *Average emotion-DOUBLE* representations);

Lex3 *Average emotion-REVERSE Representation.*

We note here, that the use of real valued weights for the words' emotion dimensions (as in eGreekSentLex) is more useful for lexicon-based vectorization compared to the binary weighting approach followed in Emolex. Also, in the case of the Emolex, the different word forms it contains often have different emotion annotations. For instance, the word "harass" is characterized as a *negative* word that expresses *anger* and *disgust*, whereas "harassing" has been simply annotated as a word that expresses *anger*. As far as the eGreekSentLex is concerned all terms included are lemmatized due to the high inflection and the morphological variety of the Greek language.

All employed representation schemes have been presented in Section 3.1.2. For the handling of Greek, words included in the documents of *MOBILE-SEN* and *MOBILE-PAR* were *lemmatized* (Section 3.1.2). This is necessary since the terms in eGreekSentLex are lemmatized. Results derived from the application of SVM on the four datasets for the aforementioned types of vectors are outlined in Table 3.

Table 3: Best achieved accuracy and corresponding SVM parameters for the lexicon-based vectors

Dataset	Vectorization Scheme	C	Accuracy (%)	F-score
MOVIES	Lex1	0.01	52.29	0.2096
	Lex2	0.1	**64.17**	0.6062
	Lex3	10	60.20	0.5775
IMDB	Lex1	0.1	82.70	0.8300
	Lex2	0.1	**83.00**	0.8300
	Lex3	0.1	70.80	0.7100
MOBILE-SEN	Lex1	0.1	72.79	0.6954
	Lex2	1	**73.32**	0.7000
	Lex3	1	63.39	0.6339
MOBILE-PAR	Lex1	10	76.70	0.8570
	Lex2	100	**78.10**	0.8680
	Lex3	10	58.10	0.6950

Comparing the three vectorization approaches, it appears that Lex2 is the most suitable for the sentiment classification task, with Lex1 being a good alternative when the vector dimensionality is an issue. Lex1 always has less features than Lex2 (half its features in the most extreme case). This highlights the importance of having the individual emotional words in the lexicon-based document representation rather than the averaged emotion dimensions (as in Lex3).

4.2.2. Experiments on Word Embedding-based Vectors

We examined three cases for deriving the word embedding-based vectors. The first involved training a Word2Vec model on the given training data (resulting in *MOB-GR*, *NIOSTO-GR* and *MOV-EN* models). The second used a wider, more generic corpus for generating the vector space (represented by the *WIKI-GR* and *GOOGLE-EN*[18]). The last approach is a variation of the second, according to which the derived *WIKI-GR* model is updated based on the two sets of Greek training data, in an effort to better adapt to the specific domain (resulting in the "updated" *WIKI-MOB-GR* and *WIKI-NIOSTO-GR* models, respectively). This approach was not feasible for the English dataset since we used an existing, non-updatable, vector space model (*GOOGLE-EN*). The characteristics and training details for all models are presented in Table 4.

After generating the Word2Vec models, we derived vectors for all documents in the training and testing set of each dataset (as described in Section 3.2.2) and applied the *model building* and *sentiment prediction* phases, accordingly. Table 5 presents the best classification accuracy achieved for the different Word2Vec models and datasets, along with the parameterization that led to the corresponding result. As one might expect, in most cases, training the Word2Vec model based on the respective corpus leads to improved overall accuracy.

[18]The *GOOGLE-EN* model was downloaded from: `http://code.google.com/p/word2vec/`. All other vector space models were trained based on the Word2Vec implementation provided in the gensim library (`https://radimrehurek.com/gensim/models/word2vec.html`).

Table 4: Word2Vec models for the generation of word embedding-based vectors

Model	Corpus used	Language	Window	Word Freq.	Words	Dimensions
GOOGLE-EN	Google News (100B words)	English	5	5	3M	300
WIKI-GR	Greek Wikipedia (2M sentences)	Greek	5	10	144,260	300
MOV-EN	*MOVIES*-Train (9,596 sentences)	English	5	5	4,101	200
MOB-GR	*MOBILE-SEN*-Train (2,520 sentences)	Greek	5	5	736	100
NIOSTO-GR	*MOBILE-PAR*-Train (6,854 sentences)	Greek	5	5	2,052	100
WIKI-MOB-GR	Greek Wikipedia *MOBILE-SEN*-Train	Greek	5	10	144,260	300
WIKI-NIOSTO-GR	Greek Wikipedia *MOBILE-PAR*-Train	Greek	5	10	144,260	300

Table 5: Best achieved accuracy and corresponding SVM parameters for each set of word-embedding based vectors.

Dataset	Corpus Used	C	Accuracy (%)	F-score
MOVIES	*MOV-EN*	0.1	**73.96**	0.7343
	GOOGLE-EN	0.1	64.32	0.6493
IMDB	*IMDB-EN*	1	**86.90**	0.8700
	GOOGLE-EN	1	86.10	0.8600
MOBILE-SEN	*MOB-GR*	1	**71.79**	0.6882
	WIKI-GR	1	69.79	0.6814
	WIKI-MOB-GR	1	70.89	0.6968
MOBILE-PAR	*NIOSTO-GR*	10	81.20	0.8890
	WIKI-GR	100	73.40	0.8360
	WIKI-NIOSTO-GR	10	**82.40**	0.8970

4.2.3. Performance Evaluation of Hybrid Vectors

Next, we evaluated the hybrid vectorization approach based on several combinations of lexicon-based and word embedding-based vectors on all four datasets, as summarized in Table 6. In this table, the *W2V* notation corresponds to the word embedding-based features derived, based on the *GOOGLE-EN* and *WIKI-GR* Word2Vec models, for the English and Greek datasets accordingly. In the hybrid approach we did not consider the Word2Vec models custom trained on the corpora (*MOV-EN*, *IMDB-EN*, *MOB-GR* and *NIOSTO-GR*) since this will limit the generalization of the approach, and instead we focused on large-scale, domain-agnostic corpora (*GOOGLE-EN* and *WIKI-GR*). The **W2V** notation corresponds to those generated through the "updated" *WIKI-MOB-GR* and *WIKI-NIOSTO-GR* models.

Our cross-validation results show that the use of the hybrid approach can improve the test-set accuracy anywhere from 5.25% to 5.5% in the Greek corpora compared to the various examined lexicon-based representations, and 7.68% to 10.2% compared to the plain Word2Vec represenation (for MOBILE-SEN and MOBILE-PAR datasets, respectively). In the English corpus, the improvement of the hybrid approach is 10.32% for *MOVIES* and 5.7% for the *IMDB* to the lexicon-based representation and 10.17% to 1.6% compared to the Word2Vec representation, respectively.

In all cases, the hybrid vectorization outperforms the accuracy of the lexicon-based and Word2Vec vectors. The most effective hybrid vectorization approach is the one that combines the *W2V* (or **W2V**) features with the Lex2 lexicon-based features.

For Greek corpora the improvement of the hybrid approach compared to the plain Word2Vec representation is higher than the respective improvement to the lexicon-based the representations. The high inflection issues and the morphological variety of the Greek language, partially tackled by the lemmatization of the lexicon-based approach, might be an explanation.

4.2.4. Performance and Running time Comparative Results

Next, we compare the top-scoring approaches for *MOVIES*, *MOBILE-SEN* and *MOBILE-PAR* (in terms of accuracy) with a well-known approach that employs RAE to detect sentiment distributions at various document levels (Socher et al., 2011). For the *IMDB* dataset we compare our results with the best results provided in the literature (Mesnil et al., 2015) and (Le & Mikolov, 2014).

The results summarized in Table 7 indicate that our hybrid methodology is equivalent to RAE in terms of performance; in more detail, the obtained accuracy is 0.62% higher for the *MOVIES* dataset, while it is 0.36% lower for the *MOBILE-SEN* and 0.39% for the *MOBILE-PAR* dataset, respectively (the statistical variability of the cross validation method indicated that such small differences might not be significant, and render the models essentially equivalent with respect to performance). At the same time, the proposed methodology is much faster than RAE since its running time is 2 or even 3 orders of magnitude (for the hybrid vectors) less than the time needed by RAE. Interestingly, the proposed approach performs significantly better on the *MOBILE-PAR* dataset compared

Table 6: Best achieved accuracy and corresponding SVM parameters for the hybrid vector. W2V corresponds to either the *GOOGLE-EN* or *WIKI-GR* Word2Vec model (depending on the dataset). *W2V* represents the *WIKI-MOB-GR* or *WIKI-NIOSTO-GR* model (depending on the dataset).

Dataset	Vectorization Scheme	C	Accuracy (%)	F-score
MOVIES	W2V+Lex1	0.1	74.09	0.7358
	W2V+Lex2	0.01	**74.49**	0.7403
	W2V+Lex3	0.1	74.12	0.7374
IMDB	W2V+Lex1	0.1	**87.80**	0.8800
	W2V+Lex2	0.1	**87.80**	0.8800
	W2V+Lex3	0.1	87.10	0.8700
MOBILE-SEN	W2V+Lex1	10	74.36	0.7367
	W2V+Lex2	10	74.93	0.7394
	W2V+Lex3	10	71.43	0.7055
	W2V+Lex1	0.1	78.00	0.7732
	W2V+Lex2	1	**78.57**	0.7785
	W2V+Lex3	1	71.79	0.7093
MOBILE-PAR	W2V+Lex1	10	76.90	0.8580
	W2V+Lex2	100	**83.60**	0.9060
	W2V+Lex3	10	80.60	0.8840
	W2V+Lex1	10	79.90	0.8790
	W2V+Lex2	1	83.40	0.9030
	W2V+Lex3	10	81.80	0.8930

Table 7: Effectiveness and efficiency comparison between the three examined vectorization approaches and the RAE algorithm.

Dataset	Methodology	Accuracy (%)	Time / fold (sec) (order of magnitude)
MOVIES	RAE	73.87	10^4
	Lexicon-based	64.17	10^1
	W2V-based	73.96	10^1
	Hybrid	**74.49**	10^1
IMDB	Mesnil *et al* (2015)	**92.57**	NA
	Le & Mikolov (2014)	**92.58**	NA
	Lexicon-based	83.00	10^2
	W2V-based	86.90	10^2
	Hybrid	**87.80**	10^2
MOBILE-SEN	RAE	**78.93**	10^3
	Lexicon-based	73.32	10^1
	W2V-based	71.79	10^1
	Hybrid	**78.57**	10^1
MOBILE-PAR	RAE	**83.99**	10^4
	Agathangelou *et al* (2014)	78.05	NA
	Lexicon-based	78.10	10^1
	W2V-based	71.79	10^1
	Hybrid	**83.60**	10^1

to the (unsupervised) method presented in Agathangelou et al. (2014), since it achieves an 83.60% accuracy, compared to the best accuracy reported which was 78.05% (5.55% higher). For the *IMDB* dataset the proposed methodology fails to outperform the approaches presented by Le & Mikolov (2014) and Mesnil et al. (2015), since it achieves an 87.80% accuracy compared to 92.58% (4.78% lower). The training time of the aforementioned models is not available, but it is indicative that in Le & Mikolov (2014) a testing time of 30 mins for the *IMDB* test set using a 16 core machine, is reported. For the same dataset our methodology requires an order of 10^2 sec (per fold) for the training and prediction on a single core processor; two orders of magnitude faster.

Comparing the different vectorization approaches employed, we observe that hybrid vectors always result in a significant improvement in performance compared to their lexicon-based and the word-embedding based counterparts. In addition, the use of hybrid vectors does not seem to significantly increase the required running time for the SVM algorithm. Therefore, based on the above, the hybrid vectorization approach emerges as a good match for the proposed sentiment analysis methodology.

5. Conclusions

In this work we propose the use of a hybrid approach for the prediction of sentiment, where we combine the context-sensitive coding offered by Word2Vec with sentiment/emotion information offered by a lexicon. We hypothesize that terms' semantic and syntactic relationships, as captured by the Word2Vec representation, or term presence/absence, as captured by a Bag-of-Words representation, are not sufficient for the task of sentiment analysis since they do not carry sentiment information. Thus, the addition of such a lexicon could offer considerable benefits. The resulting hybrid representations are then used as inputs for the supervised training of a classifier that the user chooses. There is flexibility in the choice of the classifier although our experiments indicated that the SVM model with a linear kernel is giving the best results in terms of efficiency considering both the accuracy and the process time (only results with this model have been presented here for brevity). We tested our hypothesis using four different text corpora in Greek and English, along with different coding schemes and different classifier models.

The proposed methodology is not the state-of-the-art as far as the accuracy, especially in the English language, is concerned. On the other hand state-of-the-art approaches, most of the time, are computationally costly and their performance is tested only on a single language. The proposed methodology is simple, fast and flexible and can be applied for any language (customized properly). It provides results of high accuracy in the two languages tested (state-of-the-art for the Greek language). It requires minimal computational resources, thus, it is computationally cheap and might have impact in realistic cases. For example, big data sentiment analysis with limited computational resources, or the design of a sentiment analysis standalone software application.

In the future we aim to conduct experiments towards the following directions, in order to showcase the flexibility and the computational efficiency of the methodology, and further improve its performance: (a) explore new topics beyond online customer reviews, (b) consider another high inflection language, (c) consider other emotional lexicons for the English language (since Emolex can be attributed with some inefficiencies), (d) apply the methodology into big data sentiment analysis (e.g. Twitter).

Acknowledgments

This work has been supported by mSensis S.A. and the Hellenic General Secretariat for Research and Technology (GSRT)-Programme for the Development of Industrial Research and Technology-PAVET, *Deep Learning Methods in Sentiment Analysis* (Ref. No. 1493-BET-2013). The ownership of all possible future IPRs, related in any way with this work, belong solely to mSensis S.A. as it is stated to the respective contract between mSensis S.A. and GSRT. Authors and their academic affiliations do not claim ownership to any possible future IPRs that might be related in any way with this work.

References

Agathangelou, P., Katakis, I., Kokkoras, F., & Ntonas, K. (2014). Mining domain-specific dictionaries of opinion words. In *Web Information Systems Engineering–WISE 2014* (pp. 47–62).

Carrillo-de Albornoz, J., & Plaza, L. (2013). An emotion-based model of negation, intensifiers, and modality for polarity and intensity classification. *Journal of the American Society for Information Science and Technology*, *64*, 1618–1633.

Bollen, J., Mao, H., & Zeng, X. (2011). Twitter mood predicts the stock market. *Journal of Computational Science*, *2*, 1–8.

Chang, C.-C., & Lin, C.-J. (2011). Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *2*, 27.

Chatzakou, D., Koutsonikola, V., Vakali, A., & Kafetsios, K. (2013). Microblogging content analysis via emotionally-driven clustering. In *Affective Computing and Intelligent Interaction (ACII), 2013 Humaine Association Conference on* (pp. 375–380).

Chatzakou, D., & Vakali, A. (2015). Harvesting opinions and emotions from social media textual resources. *IEEE Internet Computing*, (pp. 46–50).

Cui, H., Mittal, V., & Datar, M. (2006). Comparative experiments on sentiment classification for online product reviews. In *AAAI* (pp. 1265–1270).

DiGrazia, J., McKelvey, K., Bollen, J., & Rojas, F. (2013). More tweets, more votes: Social media as a quantitative indicator of political behavior. *PloS one*, *8*, e79449.

Ekman, P. (1992). An argument for basic emotions. *Cognition & Emotion*, *6*, 169–200.

Esuli, A., & Sebastiani, F. (2006). Sentiwordnet: A publicly available lexical resource for opinion mining. In *Proceedings of LREC* (pp. 417–422).

Ghose, A., & Ipeirotis, P. G. (2007). Designing novel review ranking systems: predicting the usefulness and impact of reviews. In *Proceedings of the Ninth International Conference on Electronic Commerce* (pp. 303–310).

Heerschop, B., Goossen, F., Hogenboom, A., Frasincar, F., Kaymak, U., & de Jong, F. (2011). Polarity analysis of texts using discourse structure. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management* (pp. 1061–1070).

Kermanidis, K. L., & Maragoudakis, M. (2013). Political sentiment analysis of tweets before and after the greek elections of may 2012. *International Journal of Social Network Mining*, *1*, 298–317.

Kotrotsios, K. (2015). *Development of tools for the automatic sentiment prediction of greek text using semi-supervised recursive autoencoders*. Master's thesis Department of Information Technology, ATEI of Thessaloniki.

Lapponi, E., Read, J., & Ovrelid, L. (2012). Representing and resolving negation for sentiment analysis. In *Data Mining Workshops (ICDMW), 2012 IEEE 12th International Conference on Data Mining Workshops* (pp. 687–692).

Le, Q. V., & Mikolov, T. (2014). Distributed representations of sentences and documents. *CoRR*, *abs/1405.4053*.

Lin, C., & He, Y. (2009). Joint sentiment/topic model for sentiment analysis. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management* (pp. 375–384).

Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., & Potts, C. (2011). Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1* (pp. 142–150).

Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S. J., & McClosky, D. (2014). The stanford corenlp natural language processing toolkit. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations* (pp. 55–60).

Medhat, W., Hassan, A., & Korashy, H. (2014). Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal*, *5*, 1093–1113.

Mesnil, G., Mikolov, T., Ranzato, M., & Bengio, Y. (2015). Ensemble of generative and discriminative techniques for sentiment analysis of movie reviews. In *Proceedings of the 2015 International Conference on Learning Representations (ICLR)* (pp. 1061–1070).

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *CoRR*, *abs/1301.3781*.

Mohammad, S. M., & Turney, P. D. (2013). Crowdsourcing a word–emotion association lexicon. *Computational Intelligence*, *29*, 436–465.

Pang, B., & Lee, L. (2005). Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics* (pp. 115–124).

Pang, B., Lee, L., & Vaithyanathan, S. (2002). Thumbs up?: Sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing-Volume 10* (pp. 79–86).

Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. In *EMNLP* (pp. 1532–1543).

Plutchik, R. (1994). *The psychology and biology of emotion*. (1st ed.). HarperCollins College Publishers.

Qiu, G., He, X., Zhang, F., Shi, Y., Bu, J., & Chen, C. (2010). Dasa: dissatisfaction-oriented advertising based on sentiment analysis. *Expert Systems with Applications*, *37*, 6182–6191.

Ren, Y., Wang, R., & Ji, D. (2016). A topic-enhanced word embedding for twitter sentiment classification. *Information Sciences*, *369*, 188–198.

Saif, H., He, Y., Fernandez, M., & Alani, H. (2016). Contextual semantics for sentiment analysis of twitter. *Information Processing and Management*, *52*, 5–19.

Severyn, A., & Moschitti, A. (2015a). Twitter sentiment analysis with deep convolutional neural networks. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '15), Santiago, Chile* (pp. 959–962).

Severyn, A., & Moschitti, A. (2015b). Unitn: Training deep convolutional neural network for twitter sentiment classification. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), Association for Computational Linguistics, Denver, Colorado* (pp. 464–469).

Socher, R., Pennington, J., Huang, E. H., Ng, A. Y., & Manning, C. D. (2011). Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (pp. 151–161).

Solakidis, G. S., Vavliakis, K. N., & Mitkas, P. A. (2014). Multilingual sentiment analysis using emoticons and keywords. In *Proceedings of the 2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)-Volume 02* (pp. 102–109).

Staiano, J., & Guerini, M. (2014). Depechemood: a lexicon for emotion analysis from crowd-annotated news. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics* (pp. 427–433).

Taboada, M., Brooke, J., Tofiloski, M., Voll, K., & Stede, M. (2011). Lexicon-based methods for sentiment analysis. *Computational Linguistics*, *37*, 267–307.

Tsakalidis, A., Papadopoulos, S., & Kompatsiaris, I. (2014). An ensemble model for cross-domain polarity classification on twitter. In *Web Information Systems Engineering–WISE 2014* (pp. 168–177).

Turney, P. D. (2002). Thumbs up or thumbs down?: Semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics* (pp. 417–424).

Wang, G., Zhang, Z., Sun, J., Yang, S., & Larson, C. A. (2015). Pos-rs: A random subspace method for sentiment classification based on part-of-speech analysis. *Information Processing and Management*, *51*, 458–479.

Wiegand, M., Balahur, A., Roth, B., Klakow, D., & Montoyo, A. (2010). A survey on the role of negation in sentiment analysis. In *Proceedings of the Workshop on Negation and Speculation in Natural Language Processing* (pp. 60–68).

Wilson, T., Wiebe, J., & Hoffmann, P. (2005). Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the Cconference on Human Language Technology and Empirical Methods in Natural Language Processing* (pp. 347–354).

Zhang, L., Ghosh, R., Dekhil, M., Hsu, M., & Liu, B. (2011). Combining lexicon-based and learning-based methods for twitter sentiment analysis. *HP Laboratories*, *HPL-2011-89*.