Expert Systems With Applications 77 (2017) 236–246

Contents lists available at ScienceDirect

Expert Systems With Applications

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

Enhancing deep learning sentiment analysis with ensemble techniques in social applications

Oscar Araque*, Ignacio Corcuera-Platas, J. Fernando Sánchez-Rada, Carlos A. Iglesias

Universidad Politécnica de Madrid, Escuela Técnica Superior de Ingenieros de Telecomunicación, Departamento de Ingeniería de Sistemas Telemáticos, Avenida Complutense 30, Madrid, Spain

ARTICLE INFO

Article history:
Received 30 June 2016
Revised 31 January 2017
Accepted 1 February 2017
Available online 3 February 2017

Keywords:
Ensemble
Deep learning
Sentiment analysis
Machine learning
Natural language processing

ABSTRACT

Deep learning techniques for Sentiment Analysis have become very popular. They provide automatic feature extraction and both richer representation capabilities and better performance than traditional feature based techniques (i.e., surface methods). Traditional surface approaches are based on complex manually extracted features, and this extraction process is a fundamental question in feature driven methods. These long-established approaches can yield strong baselines, and their predictive capabilities can be used in conjunction with the arising deep learning methods. In this paper we seek to improve the performance of deep learning techniques integrating them with traditional surface approaches based on manually extracted features. The contributions of this paper are sixfold. First, we develop a deep learning based sentiment classifier using a word embeddings model and a linear machine learning algorithm. This classifier serves as a baseline to compare to subsequent results. Second, we propose two ensemble techniques which aggregate our baseline classifier with other surface classifiers widely used in Sentiment Analysis. Third, we also propose two models for combining both surface and deep features to merge information from several sources. Fourth, we introduce a taxonomy for classifying the different models found in the literature, as well as the ones we propose. Fifth, we conduct several experiments to compare the performance of these models with the deep learning baseline. For this, we use seven public datasets that were extracted from the microblogging and movie reviews domain. Finally, as a result, a statistical study confirms that the performance of these proposed models surpasses that of our original baseline on F1-Score.

1. Introduction

The growth of user-generated content in web sites and social networks, such as Twitter, Amazon, and Trip Advisor, has led to an increasing power of social networks for expressing opinions about services, products or events, among others. This tendency, combined with the fast spreading nature of content online, has turned online opinions into a very valuable asset. In this context, many Natural Language Processing (NLP) tasks are being used in order to analyze this massive information. In particular, Sentiment Analysis (SA) is an increasingly growing task (Liu, 2015), whose goal is the classification of opinions and sentiments expressed in text, generated by a human party.

The dominant approaches in sentiment analysis are based on *machine learning* techniques (Pang, Lee, & Vaithyanathan, 2002; Read, 2005; Wang & Manning, 2012). Traditional approaches frequently use the Bag Of Words (BOW) model, where a document is mapped to a feature vector, and then classified by machine learning techniques. Although the BOW approach is simple and quite efficient, a great deal of the information from the original natural language is lost (Xia & Zong, 2010), e.g., word order is disrupted and syntactic structures are broken. Therefore, various types of features have been exploited, such as higher order *n*-grams (Pak & Paroubek, 2010). Another kind of feature that can be used is Part Of Speech (POS) tagging, which is commonly used during a syntactic analysis process, as described in Gimpel et al. (2011). Some authors refer to this kind of features as *surface* forms, as they consist in lexical and syntactical information that relies on the pattern of the text, rather than on its semantic aspect.

Some prior information about sentiment can also be used in the analysis. For instance, by adding individual word polarity to the previously described features (Pablos, Cuadros, & Rigau, 2016). This

* Corresponding author.
E-mail addresses: o.araque@upm.es (O. Araque), ignacio.cplatas@alumnos.upm.es (I. Corcuera-Platas), jfernando@dit.upm.es (J.F. Sánchez-Rada), cif@gsi.dit.upm.es (C.A. Iglesias).

http://dx.doi.org/10.1016/j.eswa.2017.02.002

prior knowledge usually takes the form of *sentiment lexicons*, which have to be gathered. Sentiment lexicons are used as a source of subjective sentiment knowledge, where this knowledge is added to the previously described features (Cambria, 2016; Kiritchenko, Zhu, & Mohammad, 2014; Melville, Gryc, & Lawrence, 2009; Nasukawa & Yi, 2003).

The use of lexicon-based techniques has a number of advantages (Taboada, Brooke, Tofiloski, Voll, & Stede, 2011). First, the linguistic content can be taken into account through mechanisms such as sentiment valence shifting (Polanyi & Zaenen, 2006) considering both intensifiers (e.g. very bad) and negations (e.g. not happy). In addition, sentiment orientation of lexical entities can be differentiated based on their characteristics. Moreover, language-dependent characteristics can be included in these approaches. Nevertheless, lexicon-based approaches have several drawbacks: the need of a lexicon that is consistent and reliable (Taboada et al., 2011), as well as the variability of opinion words across domains (Turney, 2002), contexts (Ding, Liu, & Yu, 2008) and languages (Perez-Rosas, Banea, & Mihalcea, 2012). These dependencies make it hard to maintain domain independent lexicons (Qiu, Liu, Bu, & Chen, 2009).

In general, extracting complex features from text, figuring out which features are relevant, and selecting a classification algorithm are fundamental questions in the machine learning driven methods (Agarwal, Xie, Vovsha, Rambow, & Passonneau, 2011; Sharma & Dey, 2012; Wilson, Wiebe, & Hoffmann, 2009). Traditional approaches rely on manual feature engineering, which is time consuming.

On the other hand, deep learning is a promising alternative to traditional methods. It has shown excellent performance in NLP tasks, including Sentiment Analysis (Collobert et al., 2011). The main idea of deep learning techniques is to learn complex features extracted from data with minimum external contribution (Bengio, 2009) using deep neural networks (Alpaydin, 2014). These algorithms do not need to be passed manually crafted features: they automatically learn new complex features. Nevertheless, a characteristic feature of deep learning approaches is that they need large amounts of data to perform well (Mikolov, Chen, Corrado, & Dean, 2013). Both automatic feature extraction and availability of resources are very important when comparing the traditional machine learning approach and deep learning techniques.

However, it is not clear whether the domain specialization capacity of traditional approaches can be surpassed with the generalization capacity of deep learning based models in all NLP tasks, or if it is possible to successfully combine these two techniques in a wide range of applications.

In this paper, we propose a combination of these two main sentiment analysis approaches through several ensemble models in which the information provided by many kinds of features is aggregated. In particular, this work considers an ensemble of classifiers, where several sentiment classifiers trained with different kinds of features are combined, and an ensemble of features, where the combination is made at the feature level. In order to study the complementarity of the proposed models, we use six public test datasets from two different domains: Twitter and movie reviews. Moreover, we performed a statistical study on the results of these ensemble models in comparison to a deep learning baseline we have also developed. We also present the complexity of the proposed ensemble models. Besides, we present a taxonomy that classifies the models found in the literature and the ones proposed in this work.

With our proposal we seek answers to the following questions, using the empirical results we have obtained as basis:

1. Is there a framework for characterizing existing approaches in relation to the ensemble of deep and traditional techniques in sentiment analysis?
2. Can deep learning approaches benefit from their ensemble with surface approaches?
3. How do different deep and surface ensembles compare in terms of performance?

The rest of the paper is organized as follows. Section 2 shows previous work on both ensemble techniques and deep learning approaches. Section 3 describes the proposed taxonomy for classifying ensemble methods that merge surface and deep features, whereas Section 4 addresses the proposed classifier and ensemble models. In Section 5, we describe the designed experimental setup. Experimental results are presented and analyzed in Section 6. Finally, Section 7 draws conclusions from previous results and outlines the future work.

2. Related work

In this section we offer a brief summary of the previous work in the context of ensemble methods and deep learning algorithms for Sentiment Analysis.

2.1. Ensemble methods for sentiment analysis

In the field of ensemble methods, the main idea is to combine a set of models (base classifiers) in order to obtain a more accurate and reliable model in comparison with what a single model can achieve. The methods used for building upon an ensemble approach are many, and a categorization is presented in Rokach (2005). This classification is based on two main dimensions: how predictions are combined (rule based and meta learning), and how the learning process is done (concurrent and sequential).

Regarding the first dimension, on the one hand, in *rule based* approaches predictions from the base classifiers are treated by a rule, with the aim of averaging their predictive performance. Examples of rule based ensembles are the majority voting, where the output prediction per sample is the most common class; and the weighted combination, which linearly aggregates the base classifiers predictions. On the other hand, *meta learning* techniques use predictions from component classifiers as features for a meta-learning model.

As explained in Xia, Zong, and Li (2011), weighted combinations of feature sets can be quite effective in the task of sentiment classification, since the weights of the ensemble represent the relevance of the different feature sets (e.g. n-grams, POS, etc.) to sentiment classification, instead of assigning relevance to each feature individually. The benefits of rule based ensembles were shown also in Fersini, Messina, and Pozzi (2014), where several variants of voting rules are exhaustively studied in a variety of datasets, with an emphasis on the complexity that results from the use of these approaches. In a different work, Fersini, Messina, and Pozzi (2016) have compared the majority voting rule with other approaches, using three types of subjective signals: adjectives, emoticons, emphatic expressions and expressive elongations. They report that adjectives are more impacting that the other considered signals, and that the average rule is able to ensure better performance than other types of rules. Also, in Xia et al. (2011) a meta-classifier ensemble model is evaluated, obtaining performance improvements as well. An adaptive meta-learning model is described in Aue and Gamon (2005), which offers a relatively low adaptation effort to new domains. Besides, both rule based and meta-learning ensemble models can be enriched with extra knowledge, as illustrated in Xia and Zong (2011). These authors propose the use of a number of rule based ensemble models, namely a sum rule and two weighted combination approaches trained with different loss functions. The base classifiers are trained with n-grams and POS features. These models obtain significant results for cross-domain sentiment classification.

As for the second dimension, *concurrent* models divide the original dataset into several subsets from which multiple classifiers learn in a parallel fashion, creating a classifier composite. The most popular technique that processes the sample concurrently is bagging (Rokach, 2005). Bagging intends to improve the classification by combining the predictions of classifiers built on random subsets of the original data. On the contrary, *sequential* approaches do not divide the dataset but there is an interaction between the learning steps, taking advantage from previous iterations of the learning process to improve the quality of the global classifier. An interesting sequential approach is boosting, which consists in repeatedly training low-performance classifiers on different training data. The classifiers trained in this manner are then combined into a single classifier that can achieve better performance than the component classifiers.

An example of bagging performance in the sentiment analysis task can be found in Sehgal and Song (2007), where bagging and other classification algorithms are used to show that the sentiment evolution and the stock value trend are closely related. Fersini et al. (2014) also show several experimental results in relation to the bagging techniques, attending also to the associated model complexity. Moreover, some authors have shown that bagging techniques are fairly robust to noisy data, while boosting techniques are quite sensitive (Maclin & Opitz, 1997; Melville, Shah, Mihalkova, & Mooney, 2004; Prusa, Khoshgoftaar, & Dittman, 2015). The suitability of bagging and boosting ensembles is also experimentally confirmed by Wang, Sun, Ma, Xu, and Gu (2014). This work also includes the study of a different ensemble technique, random subspace, that consists in modifying the training dataset in the feature space, rather than on the instance space. The authors stand out the better performance of random subspace in comparison with similar approaches, such as bagging and boosting. Another study (Whitehead & Yaeger, 2010) shows a comparison between bagging and boosting on a standard opinion mining task. Besides, Lin, Wang, Li, and Zhou (2015) proposes a three phase framework of multiple classifiers, where an optimal subset of classifiers is automatically chosen and trained. This framework is tested in several real-world datasets for sentiment classification.

Nevertheless, these works also show that ensemble techniques not always improve the performance in the sentiment analysis task, and that there is not a global criteria to select a certain ensemble technique.

2.2. Deep learning approaches

In the realm of Natural Language Processing much of the work in deep learning has been oriented towards methods involving learning word vector representations using neural language models (Kim, 2014). Continuous representations of words as vectors has proven to be an effective technique in many NLP tasks, including sentiment analysis (Tang, Wei, Yang et al., 2014). In this sense, *word2vec* is one of the most popular approaches that allows modeling words as vectors (Mikolov, Chen et al., 2013). Word2vec is based on the Skip-gram and CBOW models to perform the computation of the distributed representations. While CBOW aims to predict a word given its context, Skip-gram predicts the context given a word. Word2vec computes continuous vector representations of words form very large datasets. The computed word vectors retain a huge amount of syntactic and semantic regularities present in the language (Mikolov, Yih, & Zweig, 2013), expressed as relation offsets in the resulting vector space. These word-level embeddings are encoded by column vectors in an embedding matrix $W \in \mathrm{IR}^{d \times |V|}$, where $|V|$ is the size of the vocabulary. Each column $W_i \in \mathrm{IR}^d$ corresponds to the word embeddings vector of the i-th word in the vocabulary. The transformation of a word w into its word embedding vector r_w is made by using the matrix-vector product:

$$r_w = W v_w$$

where v_w is an one-hot vector of size $|V|$ which has value index at w and zero in the rest. The matrix W components are parameters to be learned, and the dimension of the word vectors d is a hyperparameter to be chosen. The vector representations computed by these techniques can result very effective when used with a traditional classifier (e.g. logistic regression) for sentiment classification, as shown by Zhang, Xu, Su, and Xu (2015). An approach based in word2vec is *doc2vec* (Le & Mikolov, 2014), that models entire sentences or documents as vectors. An additional method in representation learning is the auto-encoder, which is a type of artificial neural network applied to unsupervised learning. Auto-encoders have been used for learning new representations on a wide range of machine learning tasks, such as learning representations from distorted data, as illustrated in Chen, Weinberger, Sha, and Bengio (2014).

In deep learning for SA, an interesting approach is to augment the knowledge contained in the embedding vectors with other sources of information. This added information can be sentiment specific word embedding as in Tang, Wei, Yang et al. (2014), or as in a similar work, a concatenation of manually crafted features and these sentiment specific word embeddings (Tang, Wei, Qin, Liu, & Zhou, 2014). In the work presented by Zhang and He (2015) the feature set extracted from word embeddings is enriched with latent topic features, combining them in an ensemble scheme. They also experimentally demonstrate that these enriched representations are effective for improving the performance of polarity classification. Another approach that incorporates new information to the embeddings is described in Su, Xu, Zhang, and Xu (2014), in which deep learning is used to extract sentiment features in conjunction with semantic features. Severyn and Moschitti (2015) describe an approach where distant supervised data is used to refine the parameters of the neural network from the unsupervised neural language model. Also, a collaborative filtering algorithm can be used, as is detailed in Kim et al. (2013), where the authors add sentiment information from a small fraction of the data. In the line of adding sentiment information, in Li et al. (2015) is portrayed how a sentiment Recursive Neural Network (RNN) can be used in parallel to another neural network architecture. In general, there is a growing tendency which tries to incorporate additional information to the word embeddings created by deep learning networks. An interesting work is the one described in Vo and Zhang (2015), where both sentiment-driven and standard embeddings are used in conjunction with a variety of pooling functions, in order to extract the target-oriented sentiment of Twitter comments. Enriching the information contained in word embeddings is not the only trend in deep learning for SA. The study of the compositionality in the sentiment classification task has proven to be relevant, as shown by Socher et al. (2013). This work proposes the Recursive Neural Tensor Network (RNTN) model, and it also illustrates that RNTN outperforms previous models on both binary and fine-grained sentiment analysis. The RNTN model represents a phrase using word vectors and a parse tree, computing vectors for higher nodes in the tree using a tensor-based composition function. In relation to the ensemble schemes showed in Section 2.1, some authors (Mesnil, Mikolov, Ranzato, & Bengio, 2014) have used a geometric mean rule to combine three sentiment models: a language model approach, continuous representations of sentences and a weighted BOW. That ensemble exhibits a high performance on sentiment estimation of movie reviews, and better performance that its component classifiers.

To the best of our knowledge, a hybrid approach in which deep learning algorithms, classic feature engineering and ensemble tech-

Table 1
Proposed taxonomy for ensemble of surface and Deep features. S represents surface features, G and A stand for generic word vectors and affect word vectors, respectively. The combination of the features and/or word vectors is indicated with '+'. We consider the combination *No ensemble/S+G+A* not possible since it requires different types of features. Proposed approaches are marked with '*'.

	S	S+G	G	G+A	A	S+A	S+G+A
No ensemble	Pang et al. (2002); Read (2005) Pak and Paroubek (2010); Wang and Manning (2012) Gimpel et al. (2011); Kouloumpis, Wilson, and Moore (2011) Nasukawa and Yi (2003); Taboada et al. (2011) Melville et al. (2009); Qiu et al. (2009) Kiritchenko et al. (2014)	Su et al. (2014), Kim et al. (2013)	$\mathbf{M_{G}}^{*}$, Shirani-Mehr (2012), Collobert et al. (2011)	Severyn and Moschitti (2015)	Tang, Wei, Yang et al. (2014), Socher et al. (2013)		–
Classifier ensemble	Xia and Zong (2011); Xia et al. (2011), Aue and Gamon (2005); Fersini et al. (2016), Rokach (2005); Sehgal and Song (2007), Prusa et al. (2015); Whitehead and Yaeger (2010) Fersini et al. (2014); Lin et al. (2015) Wang et al. (2014)	$\mathbf{CEM_{SG}}^{*}$, Zhang and He (2015) Mesnil et al. (2014)					$\mathbf{CEM_{SGA}}^{*}$
Feature ensemble	Agarwal et al. (2011); Wilson et al. (2009) Xia and Zong (2010)	$\mathbf{M_{SG}}^{*}$		$\mathbf{M_{GA}}^{*}$, Li et al. (2015), Vo and Zhang (2015)		Tang, Wei, Qin et al. (2014)	$\mathbf{M_{SGA}}^{*}$

niques for sentiment analysis are used has not been thoroughly studied.

3. Ensemble taxonomyy

This section presents the proposed taxonomy for ensemble techniques applied to Sentiment Analysis in both surface and deep domains. This classification intends to summarize the work found in the literature as well as to compare these models with the ones we propose. Also, with this, we address the first question raised in Section 1 regarding how combination techniques can be classified.

The taxonomy can be expressed as combination of two different dimensions. Each dimension represents a characteristic of the studied approaches. On the one hand, one dimension considers which features are used in the model. Those features can be either surface features (which stands for *S*), generic automatic word vectors (*G*), or affect word vectors specifically trained for the sentiment analysis task (*A*). On the other hand, the other dimension attends to how the different model resources are combined. These combinations can be: using no ensemble method at all, through a ensemble of classifiers, or taking advantage of a feature ensemble. Table 1 shows a representation of this taxonomy, where the two dimensions appear as rows for the first dimension, and columns for the second dimension. We have classified all the reviewed work in this paper using the proposed taxonomy, obtaining a visual layout of the techniques that are used in each approach in relation with both ensemble methods and the combination of surface and deep features.

Regarding the dimension that tackles the ensemble techniques, in the *No ensemble* category we find the classifiers that do not make use of an ensemble technique. Under the *Classifier ensemble* category we classify the approaches that are based on ensemble techniques (Section 2.1), such as the voting rule or a meta-learning technique, to name a few. In the same manner, the *Feature ensemble* category contains the approaches that make use of feature combination techniques. The feature ensemble consists in combining different set of features into an unified set that is then fed to a learning algorithm.

As for the dimension that represents which features are used, several possibilities are represented: only surface features, generic or affect words vectors (*S, G* and *A* respectively), where only one type of feature is used. Besides, this dimension also takes into account the combination of different types of features: *S+G* (surface features combined with generic word vectors), *G+A* (generics word vectors with affect embeddings), *S+A* (surface features combined with affect word embeddings), and *S+G+A* (all three types of features combined in the same model).

These two dimensions are combined, creating a grid where the different approaches can be classified. The blank spaces in the taxonomy represent techniques that, to the extent of our knowledge, have not been studied. As such, they represent work that can be addressed in the future.

In conclusion, the introduced taxonomy provides a framework for characterizing and comparing ensemble approaches in sentiment analysis. This framework provides us with the opportunity to characterize and compare existing research works in sentiment analysis using ensemble techniques. Moreover, the framework can help us to provide guidelines to choose the most efficient and appropriate ensemble method for a specific application.

4. Sentiment analysis models

This section presents the sentiment analysis models proposed in our work. These models have been validated in the Twitter and

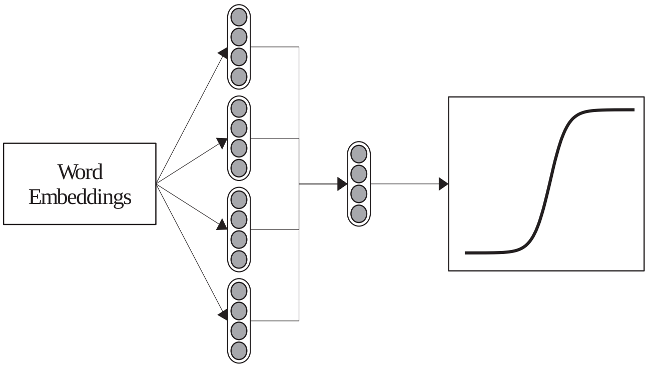

Fig. 1. Schematic representation of the baseline model, M_G. The word vectors are combined into a fixed dimension vector, and then fed to a logistic regressor, which determines the polarity of the document.

movie reviews domains (Section 5). First we describe the developed deep learning based analyzer used as baseline for the rest of the paper, and after this we detail the proposed ensemble models. These models are: ensemble of classifiers and ensemble of features. Regarding the ensemble of classifiers, we tackle two main approaches in further experiments: fixed rule and meta-learning models.

4.1. Deep learning classifier (M_G)

Generic word vectors, also denoted as pre-trained word vectors, can be captured by word embeddings techniques such as *word2vec* (Mikolov, Chen et al., 2013) and *GloVe* (Pennington, Socher, & Manning, 2014). Generic vectors are extracted in an unsupervised manner i.e., they are not trained for a specific task. These word vectors contain semantic and syntactic information, but do not enclose any specific sentiment information. Nevertheless, with the intention of exploiting the information contained in these generic word vectors, we have developed a sentiment analyzer model based on deep word embedding techniques for feature extraction, in order to compare it to other approaches in the task of Sentiment Analysis. The computed word vectors are combined into a unique vector of fixed dimension that represents the whole message. Then, this vector is fed to a logistic regression algorithm. The computation of the combined vector can be made using a set of convolutional functions, or using a embedding that transforms documents into a vector. In this way, the proposed baseline model codification is input dependent, as can be seen in Section 5. In this paper, we propose the use of word2vec for short texts, where the word vectors of the text are combined with convolutional functions; and doc2vec for long ones, representing each document with a vector. The combination of word vectors from short texts are obtained through the *min, average* and *max* convolutional functions. These functions may be combined through the concatenation of its resulting vectors. The combination of *n* of these functions produces a vector of *nd* dimensions, where *d* is the original dimension of the word vectors.

A diagram of this model is shown in Fig. 1. We refer to this model as M_G, with the G standing for generic word vectors.

4.2. Ensemble of classifiers (CEM)

Our objective is to combine the information from surface and deep features. The most straightforward method is to combine them at the classification level. In this way, we propose an ensemble model which combines classifiers trained with deep and surface features. Thus, knowledge from the two sets of features is combined, and this composition has more information than its base components. This model combines several base classifiers which make predictions from the same input data. These predictions can be subsequently used as new data for extracting a single prediction of sentiment polarity. This ensemble model aims to improve the sentiment classification performance that each base classifier can achieve individually, obtaining better performance. There are many possibilities for the combination of the base classifiers predictions that outputs a final sentiment polarity (e.g. a fixed rule or a meta learning technique). Also, any number of base classifiers can be combined into this ensemble model. A schematic diagram of this proposal is illustrated in Fig. 2. We denote this model as *CEM*, which stands for Classifier Ensemble Model. The subscript indicates the types of features its base classifiers have been trained with, like in CEM_{SG}, where the ensemble combines classifiers trained with surface features and generic word vectors.

Next, the two ensemble techniques used in this ensemble model in the experimentation section are further described.

4.2.1. Fixed rule model

This model seeks to combine the predictions from different classifiers using a simple fixed rule. Consequently, this ensemble does not need to learn from examples. The rules used in this approach can be any fixed rule used in ensemble models. In this work the rule used for the ensemble is the voting rule by majority. This rule counts the predictions of component classifiers and assigns the input to the class with most component predictions. In case of a match, a fixed class can be selected as the predicted by the model.

4.2.2. Meta classifier model

In the meta-classifier technique, the outputs of the component classifiers are treated as features for a meta-learning model. One advantage of this approach is that it can learn, i.e. adapt to different situations. As for the selection of the learning algorithm of this approach, there is no indication as of which one should be used. In this work, we select the Random Forest algorithm, as it can achieve high performance metrics in sentiment analysis (da Silva, Hruschka, & Hruschka, 2014; Zhang et al., 2011).

4.3. Ensemble of features (M_{SG} and M_{GA})

This model is proposed with the aim of combining several types of features into a unified feature set and, consequently, combine the information these features give. In this way, a learning model that learns from this unified set could achieve better performance scores that one that learns from a feature subset.

In this sense, we can distinguish two main types of ensembles of features. The first type is the ensemble of features that combines both surface and deep learning features. We address to this first model type as M_{SG}, as it combines surface features and generic word vectors. The second type consists on an ensemble of features that were completely extracted using deep learning techniques. This second type is referred as M_{GA}, combining both generic and affect word vectors. We refer to affect vectors as the result from training a set of pre-trained word vectors for a specific task, which in this case it would be SA.

Additionally, we also propose a third feature ensemble model, where all the three types of features are combined. This model, where surface features, generic word vectors and affect word vectors are combined is denoted by M_{SGA}. A diagram representing two instances of the model is shown in Fig. 2.

5. Experimental study

This section describes the experiments conducted in order to answer the questions formulated in the introduction (Section 1).

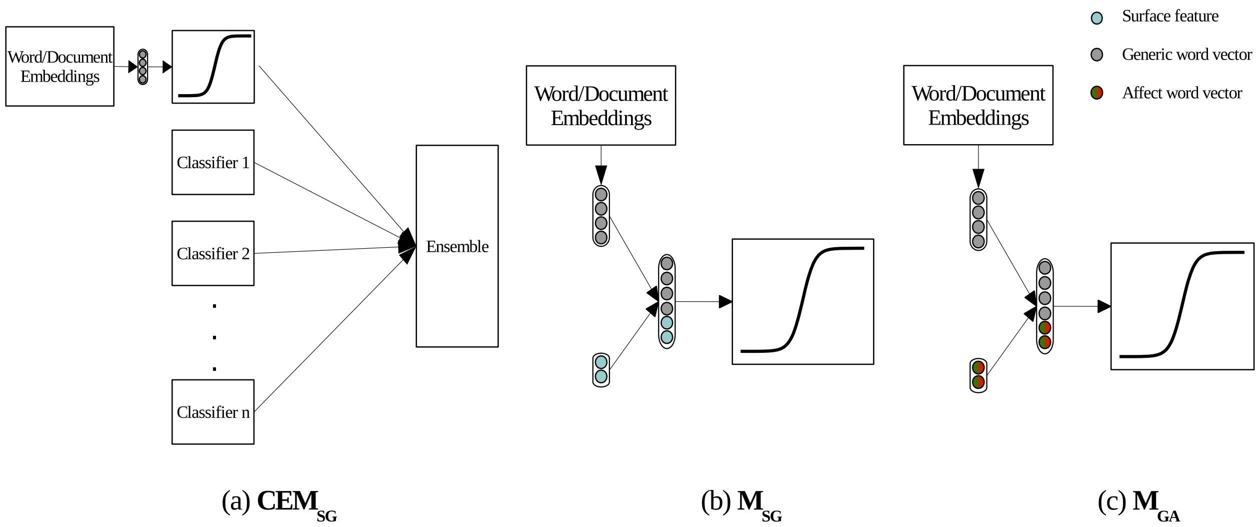

Fig. 2. Diagram of how the different classifiers and features are combined in the CEM$_{SG}$ (Classifier Ensemble Model combining surface features and generic word vectors), M$_{SG}$ and M$_{GA}$ models.

Table 2
Statistics of the SemEval2014/2014, Vader, STS-Gold and Sentiment140 datasets.

Dataset	Positive	Negative	Total	Average #words
SemEval2013	2315	861	3176	23
SemEval2014	2509	932	3441	22
Vader	2901	1299	4200	16
STS-Gold	632	1402	2034	16
Sentiment140	800,000	800,000	1,600,000	15
IMDB	25,000	25,000	50,000	255
PL04	1000	1000	2000	723

Each performance experiment is made with six different datasets, widely used by the community of Sentiment Analysis. The metric used in this work is the macro averaged F1-Score. Accuracy, Precision and Recall are also computed for all the experiments, and the interested reader can find these results in the web[1]. We also publish the computed vectors that have been used in the deep models.

These experiments (Section 6.2) are aimed to compare the performance between the deep learning baseline we have developed (M$_G$) and the proposed ensemble models. Also, some experiments (Section 6.1) are also aimed to characterize the sentiment analysis performance for each individual classifier of the CEM models. For the last purpose, we have collected several sentiment analyzers for composing a classifier ensemble.

As for the sentiment analysis of natural language, it is conducted at the message level, so it is not necessary to split the input data into sentences. The classifiers label each comment as either positive or negative.

5.1. Datasets

The datasets used for testing are *SemEval 2013, SemEval 2014* (Rosenthal, 2014), *Vader* (Hutto, 2015), *STS-Gold* (Saif, Fernandez, He, & Alani, 2013), *IMDB* (Mass et al., 2011) and *PL04* (Pang et al., 2002). Also, we use the *Sentiment 140* (Go, Bhayani, & Huang, 2009) and IMDB datasets for training and developing our deep learning baseline, M$_G$. These datasets are described next, and some statistics are summarized in Table 2.

The *SemEval 2013* test corpus is composed of English comments extracted from Twitter on a range of topics: several entities, products and events. Similarly, we have also use the *SemEval 2014* test dataset. In both *SemEval* datasets, the data is not public but must be downloaded from the source first. As some users have already deleted their comments online, we have not been able to recover the original datasets, but subsets of it. Besides, since the development dataset contains only binary targets (positive and negative), we have made an alignment processing of the *SemEval* datasets, filtering other polarity values. The obtained sizes are detailed in Table 2.

The *Vader* dataset contains 4200 tweet-like messages, originally inspired by real Twitter comments. A subset of these messages is specifically designed to test some syntactical and grammatical features that appear in the natural language. The *STS-Gold* dataset for Twitter, which has been collected as a complement for Twitter sentiment analysis evaluations processes (Saif et al., 2013).

As for the training data of our Twitter baseline model, the selected dataset is the *Sentiment 140* dataset, containing 1,600,000 Twitter messages extracted using a distant supervision approach (Go et al., 2009). The abundance of data in this dataset is very beneficial to our deep learning approach, as it requires large quantities of data to extract a fairly good model, as pointed out by Mikolov, Chen et al. (2013).

Regarding the movie reviews domain, *IMDB* contains 50,000 polarized messages, using the score of each review as a guide for the polarity value. Besides, this dataset contains 50,000 unlabeled messages that have been used for training the movie reviews baseline model. We use this dataset for the training of the movie reviews baseline model. The *PL04* dataset is a well-known dataset in this domain. For the results in this dataset, we report the 10-fold cross validation metrics using the authors' public folds, in order to make our results comparable with the ones found in the literature.

5.2. Baseline training

Due to the different characteristics of the two studied domains (Twitter and movie reviews), the vector computing process for the baseline model has been made differently. In the Twitter domain, the word vectors computed by word2vec are combined using convolutional functions. For the movie reviews domain, doc2vec is used for the combination of the word vectors. For the implementation of this model, we use the gensim library (Řehůřek & Sojka, 2010).

We found that the use of the convolutional functions in large text documents does not yield better performance than doc2vec

[1] http://gsi.dit.upm.es/~oaraque/enhancing-dl.

Table 3
Effectiveness of the convolutional functions on the Sentiment140 development dataset.

Convolutional function	F-Score
max	74.82
avg	77.53
min	74.99
max + *avg*	77.63
max + *min*	76.7
avg + *min*	77.70
max + *avg* + *min*	77.73

combinations. Hence, we performed an evaluation of these two approaches on the two development datasets. While the convolutional combinations yield a F1 score of 77.53% in the Sentiment140 dataset, they also achieve 73.66& in the IMDB dataset. When using doc2vec the F1 scores are 75.00% and 89.45% in the Sentiment140 and IMDB datasets respectively. Considering the average number of words presented in Table 2 and the difference on the performance of each approach, we use the convolutional functions for the twitter domain, where short texts are analyzed and the doc2vec technique for the movie reviews domain, which contains large documents.

Regarding the training process for the short text word embeddings, we empirically fixed the dimension of the word vectors generated to 500. We use 1,280,000 tweets randomly selected from the Sentiment 140 dataset. Once this model is extracted, we feed a logistic regression model (implementation from scikit-learn) with the vectors of each tweet and the labels from the original dataset. The movie reviews baseline has been similarly trained, with the 50,000 unsupervised documents of the IMDB dataset, setting the dimension of the document vectors to 100. The same linear model is use for the classification of the document vectors. All the performance metrics have been obtained using K-fold cross validation, with folds of 10.

With respect to the convolutional functions, we have conducted an effectiveness test of the *max, average* and *min* functions on the Sentiment140 development set. The results are shown in Table 3. As can be seen, the *avg* function is very close to the performance of the complete set of functions *max, avg* and *min*. Consequently, we select the *avg* function as the one used for further experiments, as it provides very good results compared to the rest, and it also reduces the computational complexity of the experimentation. No pooling functions are used in the movie reviews, as there is no need to combine different word vectors.

Lastly, the preprocessing of natural language, we tokenized the input data and removed punctuation, excepting the most common (‘.,!?’). We also transformed URLs, numbers and usernames (@username) into especial characters to normalize the data. The preprocessing is applied to all the texts before generating the word vectors.

5.3. Ensemble of classifiers

In order to improve the performance of the deep learning baseline, we have built an ensemble composed of this analyzer and six different sentiment classifiers. Following, a list and a brief description of each of these classifiers is shown:

- sentiment140 (Go et al., 2009). It uses Naive Bayes, Maximum Entropy and Support Vector Machines trained with unigrams, bigrams and POS features.
- Stanford CoreNLP (Manning et al., 2014) is the RNTN approach shown in Section 2.2, proposed by Socher et al. (2013).
- Sentiment WSD (Kathuria, 2015), which uses SentiWordNet (Esuli & Sebastiani, 2006), performing the sentiment estimation based on the polarities of each word.
- Vivekn (Narayanan, Arora, & Bhatia, 2013). It is based in a Naive Bayes classifier trained with word n-grams and using several techniques, such as negation handling, feature selection and laplacian smoothing.
- pattern.en (De Smedt & Daelemans, 2012) uses a Support Vector Machines algorithm fed with polarity and subjectivity values for each word, WordNet vocabulary information and POS annotation.
- TextBlob Sentiment Classifier (Loria, 2016), a modular approach which as default configuration uses a Naive Bayes classifier trained with unigram features.

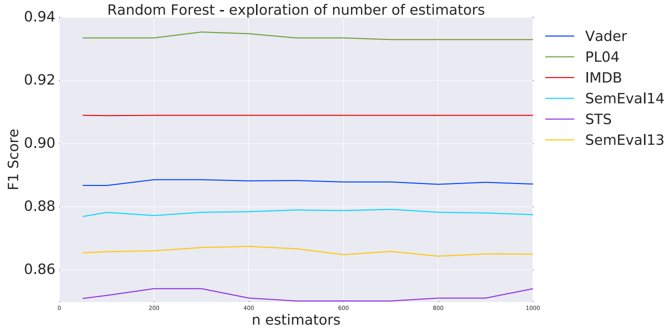

Fig. 3. Cross validation of the number of estimators on the Random Forest algorithm used for the meta-learning ensemble.

We have built ensemble classifiers using two combining techniques in the CEM model: a rule based method and a meta learning approach, both using the predictions of the classifiers composing the ensemble as features for the next step. For the meta-learning approach, we use the implementation of scikit-learn of the Random Forest algorithm. For this algorithm we have used 100 as the default number of estimators. As is shown in Fig. 3, the value of this parameter does not affect to the classification performance in the range from 50 to 1000.

Additionally, two versions of the CEM model have been implemented for the experiments. While the $\mathrm{CEM}_{\mathrm{SG}}$ combines the six aforementioned classifiers and the M_{G} model; the $\mathrm{CEM}_{\mathrm{SGA}}$ version combines the base classifiers from $\mathrm{CEM}_{\mathrm{SG}}$ with the M_{SG}, $\mathrm{M}_{\mathrm{SG+bigrams}}$ and M_{GA} models.

5.4. Ensemble of features

Based on the work by Mohammad, Kiritchenko, and Zhu (2013), we have selected the following surface features: SentiWordnet (Esuli & Sebastiani, 2006) lexicon values for each word, as well as total number of positive, neutral and negative words extracted with this lexicon; number of exclamation, interrogation and hashtags marks ‘!?#’; number of words that are all in caps and number of words that have been elongated ‘gooooood’. This feature set has been cross validated on the development sets, with the objective of obtaining the smaller surface feature set that yields the best classification performance.

With the aim of complementing the surface features, we have also explored the role of n-grams. More specifically, bigrams are used, as the introduction of unigrams and trigrams did not improve the classification performance. The ensemble of features model that includes generic word vectors, the described surface features and bigrams is represented as $\mathrm{M}_{\mathrm{SG+bigrams}}$.

As for the M_{GA} model, we use the word vectors obtained by Tang, Wei, Yang et al. (2014). More specifically, we use the vectors extracted using the $\mathrm{SSWE}_{\mathrm{r}}$ neural model. These vectors have been

Table 4
Macro averaged F-Score of all the base sentiment classifiers. TB represents the TextBlob classifier.

Dataset	sent140	CoreNLP	WSD	vivekn	pattern	TB
SemEval2013	78.92	46.95	76.18	72.14	82.50	82.51
SemEval2014	60.67	42.95	75.35	59.97	71.86	71.92
Vader	78.76	60.19	77.75	63.54	85.98	85.71
STS-Gold	75.07	59.68	75.35	69.61	82.86	68.27
IMDB	72.91	88.56	87.41	87.45	75.43	75.47
PL04	71.66	86.09	79.03	86.22	70.64	70.82

extracted for learning sentiment-specific word vectors but not general semantic information, so we use them as affect word vectors. Also, for the composition of a tweet vector, we have used the *average* function on the word vectors, as the combination of the other convolutional functions did not improve the results of the model.

6. Results

The conducted experiments show the sentiment classification performance of each base classifier separately (including our deep learning baseline) on each of the six test datasets, as well as the metrics for the ensemble of classifiers and ensembles of features. In this section, the experimental results are shown and discussed. The experimental results for the proposed models are gathered in Table 5.

6.1. Base classifiers performance

As it can be seen in Table 4, the better F-score performance is achieved by TextBlob in SemEval2013, by the WSD classifier with an important difference over TextBlob in SemEval2014; while pattern.en has a slightly better performance than the rest in the Vader dataset and has also the better performance in the STS-Gold dataset by far. The classifier with the lower performance is CoreNLP in all Twitter test sets. In contrast, in the IMDB and PL04 datasets (movie reviews), CoreNLP achieves the better performance in IMDB and the second best in PL04.

As expected, the nature of the domains has a strong impact on the performance of the base classifiers, since they have been trained in a specific domain. Some classifiers (e.g. WSD, pattern and TextBlob) are more suitable to the Twitter domain, while others (e.g. CoreNLP and vivekn) are better adapted to the review domain. In general, none of the base classifiers exhibits a high performance in all the baseline datasets.

Finally, the average F1 score performance for the base classifiers is 73.02, 63.79, 75.32, 71.81, 81.20 and 77.41% in the SemEval2013, SemEval2014, Vader, STS-Gold, IMDB and PL04 datasets respectively.

6.2. Classifiers and features classifiers performance

CEM models gather the predictions from M_G baseline and the other six base classifiers whose classification performance has been analyzed. The voting and meta-learning techniques are used as ensemble techniques. It can be seen in the Table 5 that nearly all the ensemble models surpass the baseline, as well as all the other base classifiers. In fact, the best performance is achieved in 4 out of 6 datasets by these CEM classifiers.

As for the feature ensemble models, they also push the performance further than the baseline. The $M_{SG+bigrams}$ ensemble is very close to the best metrics in almost all the test datasets. Also, it seems that $M_{SGA+bigrams}$ is suffering overfitting, as the combination of all three types of features decreases the performance when comparing to $M_{SG+bigrams}$ model. This could be due to the increase in the number of features used to train the model.

Moreover, as an additional observation, it can be seen that the better improvements are achieved by CEM_{SGA}^{Vo} and CEM_{SG}^{MeL} models, with 3.65 and 2.53% of performance gain in STS-Gold and SemeEval2013 datasets respectively, and by CEM_{SGA}^{MeL} model in the IMDB and PL04 datasets, with improvements of 1.48% and 5.77% respectively.

Although the biggest improvements have been achieved with CEM models, the feature ensemble models also improve the baseline, sometimes by a large margin. Considering this type of models, the better results are achieved by the $M_{SG+bigrams}$ model. This fact could be explained attending to fact that bigrams can successfully capture modified verbs and nouns (Wang & Manning, 2012), such as the negation.

Also, the M_{SG} model results are comparable, generally, to the best performances in the Twitter domain. Nevertheless, this model does not yield such results in the movie reviews domain. This result indicates that combining word vectors through convolutional functions in long texts does not lead to high sentiment classification performances. We can conclude that the transformation of the convolutional functions on the sentiment signals that are contained in the affect word vectors is retained when applied to short texts, but lost in long texts.

Attending to the difference of performance between the $M_{SGA+bigrams}$ and the CEM_{SGA}^{Vo} and CEM_{SGA}^{MeL} models, we see that the same types of features do not yield the same result. We make the assumption that dividing the features into smaller sets, as it is done in the ensemble models, benefits the classification performance. The division is made at the classifier level, since these ensemble models combine the predictions of classifiers trained with features (e.g. surface, generic and affect). Considering that the whole set SGA (including bigrams)is a complex collection of features, and based on the experimental results, the assumption is that this division of features prevents overfitting. These results are in line with those by Alpaydin (2014) and Xia et al. (2011). This shows that when dividing a complex set of features into simpler subsets, an ensemble can yield better performance.

6.3. Statistical analysis

In order to compare the different proposed models in this work, a statistical test has been applied on the experimental results. Concretely, the Friedman test with the corresponding Bonferroni-Dun

Table 5
Macro averaged F-Score of the proposed sentiment classifiers. The last row shows the Friedman rank. In bold the best classifier for each dataset, and in the last row the best classifier attending to the Bonferroni-Dunn test.

Dataset	M_G	CEM_{SG}^{Vo}	CEM_{SG}^{MeL}	M_{SG}	$M_{SG+bigrams}$	M_{GA}	$M_{SGA+bigrams}$	CEM_{SGA}^{Vo}	CEM_{SGA}^{MeL}
SemEval2013	85.34	87.78	**87.87**	86.36	86.53	87.54	86.26	86.26	86.97
SemEval2014	86.16	84.16	87.63	87.03	**88.19**	88.05	86.94	85.90	88.07
Vader	87.71	87.92	88.85	88.07	88.93	88.89	88.89	**89.52**	89.48
STS-Gold	83.43	83.52	84.56	84.73	**89.24**	85.27	85.26	87.08	85.59
IMDB	89.45	84.06	89.68	89.58	90.41	89.50	90.41	89.92	**90.93**
PL04	88.72	86.48	93.65	88.39	94.33	88.67	86.76	93.87	**94.49**
Friedman rank	7.83	7.5	4.5	6.17	**2.67**	4.33	5.58	4.25	**2.17**

post-hoc test, that are described by Demšar (2006). These tests are specially oriented to the comparison of several classifiers on multiple data sets.

Friedman's test is based on the rank of each algorithm in each dataset, where the best performing algorithm gets the rank of 1, the second best gets rank 2, etc. Ties are resolved by averaging their ranks. r_j^i is the rank of the jth of the k algorithms and on the i-th of N datasets. Friedman test uses the comparison of average ranks $R_j = \frac{1}{N}\sum_i r_i^j$, and states that under the null-hypothesis (all the algorithms are equal so their ranks R_j are equal) the Friedman statistic, with $k-1$ degrees of freedom, is:

$$\chi_F^2 = \frac{12N}{k(k+1)}\left(\sum_j R_j^2 - \frac{k(k+1)^2}{4}\right)$$

Nevertheless, Demšar (2006) shows that there is a better statistic that is distributed according to the F-distribution, and has $k-1$ and $(k-1)(N-1)$ degrees of freedom:

$$F_F = \frac{(N-1)\chi_F^2}{N(k-1)-\chi_F^2}$$

If the null-hypothesis of the Friedman test is rejected, post-hoc tests can be conducted. In this work we employ the Bonferroni-Dunn test, as it allows to compare the results of several algorithms to a baseline. In this case, all the proposed models are compared against $\mathrm{M_G}$. This test can be computed through comparing the critical difference (CD) with a series of critical values (q_α), which Demšar (2006) summarizes. The critical difference can be computed as:

$$CD = q_\alpha\sqrt{\frac{k(k+1)}{6N}}$$

For the computation of both tests, the ranks have been obtained. The average ranks (R_j) are showed in Table 5. The α values is set to 0.05 for the following calculations. With those averages, $\chi_F^2 = 24.56$, $F_F = 5.24$, and the critical value $F(k-1,(k-1)(N-1)) = 2.18$. As $F_F > F(8, 40)$, the null-hypothesis is rejected and the post-hoc test can be conducted. According with this, the critical difference is 4.31. Following, the difference between the average ranks of the baseline and the jth model is compared to the CD and, if greater, we can conclude that the jth algorithm performs significantly better than the baseline.

Friedman's test has pointed the $\mathrm{CEM}_{\mathrm{SGA}}^{\mathrm{MeL}}$ and the $\mathrm{M}_{\mathrm{SG+bigrams}}$ models as the two best classification models, followed by the $\mathrm{CEM}_{\mathrm{SGA}}^{\mathrm{Vo}}$ and $\mathrm{M_{GA}}$ models. Besides, the Bonferroni-Dunn test points out that $\mathrm{CEM}_{\mathrm{SGA}}^{\mathrm{MeL}}$ and $\mathrm{M}_{\mathrm{SG+bigrams}}$ models perform significantly better that the baseline. These results indicate that the hypothesis suggested in question 2 is supported, since the combination of different sources of information improves the performance of sentiment analysis. As for the rest of the models, it is not possible to reach a conclusion with the current data.

On top of this, an interesting result of these experiments is that the performance of the meta learning approach is higher than that of the fixed rule scheme. While the meta learning ensemble with all types of features (SGA + bigrams) is significantly better than the baseline, the voting model is not. This could be caused by the learning capabilities of the meta-classifier technique, feature that the fixed ensemble methods like voting rule do not have.

6.4. Computational complexity

One possible drawback of ensemble approaches is their higher cost in terms of computational resources. With the aim of analyzing the efficiency of the proposed models, the computational cost is presented. The results for this cost analysis are studied at train time, since the costs in the test phase do not result relevant for this analysis. All these measurements were made in a Intel Xeon with 12 cores available and a memory friendly environment.

Table 6
Computational complexity of the word embeddings approaches, in both model training and vector computation.

Word Embeddings approach		Sentiment140	IMDB
word2vec	Train model	109.7 s	96.1 s
	Compute vectors	152.9 s	80.4 s
doc2vec	Train model	7 h	2 h
	Compute vectors	8.5 s	6.4 s
SSWE	Train model	–	25 d
	Compute vectors	87.5 s	179.2 s

Table 7
Computational complexity of the proposed models in training time.

Model	Sentiment140	IMDB
$\mathrm{M_G}$	12.7 s	0.9 s
$\mathrm{M_{SG}}$	13.7 s	0.9 s
$\mathrm{M_{SG+bigrams}}$	977.5 s	37.2 s
$\mathrm{M_{GA}}$	12.9 s	1 s
$\mathrm{M_{SGA+bigrams}}$	977.8 s	37.4 s

In relation to the training and computation of the word embeddings approaches, Table 6 presents the associated computational cost. It can be seen that word2vec is the lighter model at train time by a large margin. Also, the implementation of SSWE largely increases the computational complexity. We believe that implementing this model for a GPU environment can have a great impact on the time performance of the SSWE training. Please note that the SSWE trained model for Twitter is available for research, and so the training using the Sentiment140 dataset has not been performed. Besides, we can see that computing the pooling functions on the word vectors increases the complexity, as can be seen by comparing with the doc2vec approach.

Combining different sets of features increases the computational complexity, as Table 7 shows. The largest increment can be found in the $\mathrm{M_{SG+bigrams}}$ and $\mathrm{M_{SGA+bigrams}}$ models, which use bigrams in the learning process. In this way, it can be seen that using feature ensemble with bigrams and other sets of features leads to the addition of complexity to the model. The difference of training times between the Sentiment140 and IMDB datasets is due to their number of instances, being larger in the first.

Finally, the CEM models do not introduce a relevant complexity to the model at training time. The ensemble of classifiers based on the voting scheme do hardly introduce a cost to the computation, as there is no learning process in this case. For the meta-learning scheme, the maximum time taken in the meta learning process is 1.5 s, with no significant difference between the training data.

7. Conclusions and future work

This paper proposes several models where classic hand-crafted features are combined with automatically extracted embedding features, as well as the ensemble of analyzers that learn from these varied features. In order to classify these different approaches, we propose a taxonomy of ensembles of classifiers and features that is based on two dimensions. Furthermore, with the aim of evaluating the proposed models, a deep learning baseline is defined, and the classification performances of several ensembles are compared to the performance of the baseline. With the intention of conducting a comparative experimental study, six public datasets are used for the evaluation of all the models, as well as six public sentiment classifiers. Finally, we conduct an statistical analysis in order to empirically verify that combining information from varied features and/or analyzers is an adequate way to surpass the sentiment classification performance.

There were three main research questions that drove this work. The first question was whether there is a framework to characterize existing approaches in relation to the ensemble of traditional and deep techniques in sentiment analysis. To the best of our knowledge, our proposal of a taxonomy and the resulting implementations is the first work to tackle this problem for sentiment analysis.

The second question was whether the sentiment analysis performance of a deep classifier can be leveraged when using the proposed ensemble of classifiers and features models. Observing the scores table and the Friedman ranks (Table 5), we see that the proposed models generally improve the performance of the baseline. This indicates that the combination of information from diverse sources such as surface features, generic and affect word vectors effectively improves the classifier's results in sentiment analysis tasks.

Lastly, we raised the concern of which of the proposed models stand out in the improvement of the deep sentiment analysis performance. In this regard, the statistical results point out the CEM^{MeL}_{SGA} and $M_{SG+bigrams}$ models as the best performing alternatives. As expected, these models effectively combine different sources of sentiment information, resulting in a significant improvement with respect to the baseline. We remark the $M_{SG+bigrams}$ model, as it does not involve an ensemble of many classifiers, but only a classifier that is trained with an ensemble of deep and surface features.

To summarize, this work takes advantage of the ensemble of existing traditional sentiment classifiers, as well as the combination of generic, sentiment-trained word embeddings and manually crafted features. Nevertheless, Considering the results of this work, we believe that a possible line of work would be applying these models to the task of aspect based sentiment analysis, with the hope of improving the classification performance. Furthermore, we intend to extend the domain of the proposed models to other languages and even paradigms, like Emotion analysis.

Acknowledgements

This research work is supported by the EC through the H2020 project MixedEmotions (Grant Agreement no: 141111), the Spanish Ministry of Economy under the R&D project Semola (TEC2015-68284-R) and the project EmoSpaces (RTC-2016-5053-7); by ITEA3 project SOMEDI (15011); and by MOSI-AGIL-CM (grant P2013/ICE-3019, co-funded by EU Structural Funds FSE and FEDER).

References

Agarwal, A., Xie, B., Vovsha, I., Rambow, O., & Passonneau, R. (2011). Sentiment analysis of twitter data. In *Proceedings of the workshop on languages in social media* (pp. 30–38). Association for Computational Linguistics.

Alpaydin, E. (2014). *Introduction to machine learning*. MIT press.

Aue, A., & Gamon, M. (2005). Customizing sentiment classifiers to new domains: A case study. In *Proceedings of recent advances in natural language processing (RANLP): vol. 1*. 2–1

Bengio, Y. (2009). Learning deep architectures for ai. *Foundations and Trends® in Machine Learning, 2*, 1–127.

Cambria, E. (2016). Affective computing and sentiment analysis. *IEEE Intelligent Systems, 31*, 102–107.

Chen, M., Weinberger, K. Q., Sha, F., & Bengio, Y. (2014). Marginalized denoising auto-encoders for nonlinear representations. In *ICML* (pp. 1476–1484).

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011). Natural language processing (almost) from scratch. *The Journal of Machine Learning Research, 12*, 2493–2537.

De Smedt, T., & Daelemans, W. (2012). Pattern for python. *The Journal of Machine Learning Research, 13*, 2063–2067.

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research, 7*, 1–30.

Ding, X., Liu, B., & Yu, P. S. (2008). A holistic lexicon-based approach to opinion mining. In *Proceedings of the 2008 international conference on web search and data mining WSDM '08* (pp. 231–240). New York, NY, USA: ACM. doi:10.1145/1341531.1341561.

Esuli, A., & Sebastiani, F. (2006). Sentiwordnet: A publicly available lexical resource for opinion mining. In *Proceedings of LREC: 6* (pp. 417–422). Citeseer.

Fersini, E., Messina, E., & Pozzi, F. (2014). Sentiment analysis: Bayesian ensemble learning. *Decision Support Systems, 68*, 26–38. doi:10.1016/j.dss.2014.10.004. http://www.sciencedirect.com/science/article/pii/S0167923614002498.

Fersini, E., Messina, E., & Pozzi, F. (2016). Expressive signals in social media languages to improve polarity detection. *Information Processing & Management, 52*, 20–35.

Gimpel, K., Schneider, N., O'Connor, B., Das, D., Mills, D., Eisenstein, J., et al. (2011). Part-of-speech tagging for twitter: Annotation, features, and experiments. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies: Short papers - volume 2 HLT '11* (pp. 42–47). Stroudsburg, PA, USA: Association for Computational Linguistics. http://dl.acm.org/citation.cfm?id=2002736.2002747.

Go, A., Bhayani, R., & Huang, L. (2009). Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford, 1*, 12.

Hutto, C. (2015). *Vader sentiment github repository*. https://github.com/cjhutto/vaderSentiment. Accessed on May 30, 2016.

Kathuria, P. (2015). *Sentiment wsd github repository*. https://github.com/kevincobain2000/sentiment_classifier/. Accessed on May 30, 2016.

Kim, J., Yoo, J.-B., Lim, H., Qiu, H., Kozareva, Z., & Galstyan, A. (2013). Sentiment prediction using collaborative filtering.. *ICWSM*.

Kim, Y. (2014). *Convolutional neural networks for sentence classification*. arXiv preprint arXiv:1408.5882.

Kiritchenko, S., Zhu, X., & Mohammad, S. M. (2014). Sentiment analysis of short informal texts. *Journal of Artificial Intelligence Research*, 723–762.

Kouloumpis, E., Wilson, T., & Moore, J. D. (2011). Twitter sentiment analysis: The good the bad and the omg!. *Icwsm, 11*, 538–541.

Le, Q. V., & Mikolov, T. (2014). Distributed representations of sentences and documents.. In *ICML: vol. 14* (pp. 1188–1196).

Li, C., Xu, B., Wu, G., He, S., Tian, G., & Zhou, Y. (2015). Parallel recursive deep model for sentiment analysis. In T. Cao, E.-P. Lim, Z.-H. Zhou, T.-B. Ho, D. Cheung, & H. Motoda (Eds.), *Advances in knowledge discovery and data mining*. In *Lecture Notes in Computer Science: Vol. 9078* (pp. 15–26). Springer International Publishing. doi:10.1007/978-3-319-18032-8_2. http://dx.doi.org/10.1007/978-3-319-18032-8_2.

Lin, Y., Wang, X., Li, Y., & Zhou, A. (2015). Integrating the optimal classifier set for sentiment analysis. *Social Network Analysis and Mining, 5*, 1–13.

Liu, B. (2015). *Sentiment analysis: Mining opinions, sentiments, and emotions*. Cambridge University Press.

Loria, S. (2016). *Textblob documentation page*. https://textblob.readthedocs.org/en/dev/index.html. Accessed on May 30, 2016.

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). Association for Computational Linguistics.

Maclin, R., & Opitz, D. (1997). An empirical evaluation of bagging and boosting. *AAAI/IAAI, 1997*, 546–551.

Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S. J., & McClosky, D. (2014). The stanford CoreNLP natural language processing toolkit. In *Association for computational linguistics (ACL) system demonstrations* (pp. 55–60). http://www.aclweb.org/anthology/P/P14/P14-5010.

Melville, P., Gryc, W., & Lawrence, R. D. (2009). Sentiment analysis of blogs by combining lexical knowledge with text classification. In *Proceedings of the 15th ACM SIGKDD international conference on knowledge discovery and data mining KDD '09* (pp. 1275–1284). New York, NY, USA: ACM. doi:10.1145/1557019.1557156. http://doi.acm.org/10.1145/1557019.1557156.

Melville, P., Shah, N., Mihalkova, L., & Mooney, R. J. (2004). Experiments on ensembles with missing and noisy data. In F. Roli, J. Kittler, & T. Windeatt (Eds.), *Multiple classifier systems: 5th international workshop, MCS 2004, Cagliari, Italy, june 9–11, 2004. proceedings* (pp. 293–302). Berlin, Heidelberg: Springer Berlin Heidelberg. doi:10.1007/978-3-540-25966-4_29. http://dx.doi.org/10.1007/978-3-540-25966-4_29.

Mesnil, G., Mikolov, T., Ranzato, M., & Bengio, Y. (2014). Ensemble of generative and discriminative techniques for sentiment analysis of movie reviews. *arXiv preprint arXiv:1412.5335*.

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

Mikolov, T., Yih, W.-t., & Zweig, G. (2013). Linguistic regularities in continuous space word representations.. In *HLT-NAACL* (pp. 746–751).

Mohammad, S. M., Kiritchenko, S., & Zhu, X. (2013). Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets. *CoRR, abs/1308.6242*. http://arxiv.org/abs/1308.6242.

Narayanan, V., Arora, I., & Bhatia, A. (2013). Fast and accurate sentiment classification using an enhanced naive bayes model. *CoRR, abs/1305.6143*. http://arxiv.org/abs/1305.6143.

Nasukawa, T., & Yi, J. (2003). Sentiment analysis: Capturing favorability using natural language processing. In *Proceedings of the 2nd international conference on Knowledge capture* (pp. 70–77). ACM.

Pablos, A. G., Cuadros, M., & Rigau, G. (2016). A comparison of domain-based word polarity estimation using different word embeddings. In N. C. C. Chair), K. Choukri, T. Declerck, S. Goggi, M. Grobelnik, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, & S. Piperidis (Eds.), *Proceedings of the tenth international conference on language resources and evaluation (LREC 2016)*. Paris, France: European Language Resources Association (ELRA).

Pak, A., & Paroubek, P. (2010). Twitter as a corpus for sentiment analysis and opinion mining.. In *LREc: vol. 10* (pp. 1320–1326).

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). Association for Computational Linguistics.

Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation.. In *EMNLP: vol. 14* (pp. 1532–1543).

Perez-Rosas, V., Banea, C., & Mihalcea, R. (2012). Learning sentiment lexicons in spanish.. In *LREC: vol. 12* (p. 73).

Polanyi, L., & Zaenen, A. (2006). Computing attitude and affect in text: Theory and applications. In *chapter Contextual Valence Shifters* (pp. 1–10)). Dordrecht: Springer Netherlands. http://dx.doi.org/10.1007/1-4020-4102-0_1.

Prusa, J., Khoshgoftaar, T., & Dittman, D. (2015). Using ensemble learners to improve classifier performance on tweet sentiment data. In *Information reuse and integration (IRI), 2015 IEEE international conference on* (pp. 252–257). doi:10.1109/IRI.2015.49.

Qiu, G., Liu, B., Bu, J., & Chen, C. (2009). Expanding domain sentiment lexicon through double propagation.. In *IJCAI: vol. 9* (pp. 1199–1204).

Read, J. (2005). Using emoticons to reduce dependency in machine learning techniques for sentiment classification. In *Proceedings of the ACL student research workshop* (pp. 43–48). Association for Computational Linguistics.

Řehůřek, R., & Sojka, P. (2010). Software framework for topic modelling with large corpora. In *Proceedings of the LREC 2010 workshop on new challenges for NLP frameworks* (pp. 45–50). Valletta, Malta: ELRA. http://is.muni.cz/publication/884893/en.

Rokach, L. (2005). Ensemble methods for classifiers. In O. Maimon, & L. Rokach (Eds.), *Data mining and knowledge discovery handbook* (pp. 957–980). Springer US. http://dx.doi.org/10.1007/0-387-25465-X_45.

Rosenthal, S. (2014). *Semeval 2014 task 9 description*. http://alt.qcri.org/semeval2014/task9/. Accessed on May 30, 2016.

Saif, H., Fernandez, M., He, Y., & Alani, H. (2013). *Evaluation datasets for twitter sentiment analysis: a survey and a new dataset, the sts-gold*.

Sehgal, V., & Song, C. (2007). Sops: Stock prediction using web sentiment. In *Data mining workshops, 2007. ICDM workshops 2007. seventh IEEE international conference on* (pp. 21–26). doi:10.1109/ICDMW.2007.100.

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

Sharma, A., & Dey, S. (2012). A comparative study of feature selection and machine learning techniques for sentiment analysis. In *Proceedings of the 2012 ACM Research in Applied Computation Symposium* (pp. 1–7). ACM.

Shirani-Mehr, H. (2012). *Applications of deep learning to sentiment analysis of movie reviews*.

da Silva, N. F., Hruschka, E. R., & Hruschka, E. R. (2014). Tweet sentiment analysis with classifier ensembles. *Decision Support Systems, 66*, 170–179.

Socher, R., Perelygin, A., Wu, J. Y., Chuang, J., Manning, C. D., & Ng, A. Y. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the conference on empirical methods in natural language processing (EMNLP): 1631* (p. 1642). Citeseer.

Su, Z., Xu, H., Zhang, D., & Xu, Y. (2014). Chinese sentiment classification using a neural network tool word2vec. In *Multisensor fusion and information integration for intelligent systems (MFI), 2014 international conference on* (pp. 1–6). doi:10.1109/MFI.2014.6997687.

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

Tang, D., Wei, F., Qin, B., Liu, T., & Zhou, M. (2014). Coooolll: A deep learning system for twitter sentiment classification. In *Proceedings of the 8th international workshop on semantic evaluation (semeval 2014)* (pp. 208–212).

Tang, D., Wei, F., Yang, N., Zhou, M., Liu, T., & Qin, B. (2014). Learning sentiment-specific word embedding for twitter sentiment classification.. In *ACL (1)* (pp. 1555–1565).

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). Association for Computational Linguistics.

Vo, D.-T., & Zhang, Y. (2015). Target-dependent twitter sentiment classification with rich automatic features. In *Proceedings of the twenty-fourth international joint conference on artificial intelligence (IJCAI 2015)* (pp. 1347–1353).

Wang, G., Sun, J., Ma, J., Xu, K., & Gu, J. (2014). Sentiment classification: The contribution of ensemble learning. *Decision support systems, 57*, 77–93.

Wang, S., & Manning, C. D. (2012). Baselines and bigrams: Simple, good sentiment and topic classification. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2* (pp. 90–94). Association for Computational Linguistics.

Whitehead, M., & Yaeger, L. (2010). Sentiment mining using ensemble classification models. In *Innovations and advances in computer sciences and engineering* (pp. 509–514). Springer.

Wilson, T., Wiebe, J., & Hoffmann, P. (2009). Recognizing contextual polarity: An exploration of features for phrase-level sentiment analysis. *Computational linguistics, 35*, 399–433.

Xia, R., & Zong, C. (2010). Exploring the use of word relation features for sentiment classification. In *Proceedings of the 23rd international conference on computational linguistics: Posters COLING '10* (pp. 1336–1344). Stroudsburg, PA, USA: Association for Computational Linguistics. http://dl.acm.org/citation.cfm?id=1944566.1944719.

Xia, R., & Zong, C. (2011). A pos-based ensemble model for cross-domain sentiment classification.. In *IJCNLP* (pp. 614–622). Citeseer.

Xia, R., Zong, C., & Li, S. (2011). Ensemble of feature sets and classification algorithms for sentiment classification. *Information Sciences, 181*, 1138–1152. doi:10.1016/j.ins.2010.11.023. http://www.sciencedirect.com/science/article/pii/S0020025510005682.

Zhang, D., Xu, H., Su, Z., & Xu, Y. (2015). Chinese comments sentiment classification based on word2vec and svm perf. *Expert Systems with Applications, 42*, 1857–1863.

Zhang, K., Cheng, Y., Xie, Y., Honbo, D., Agrawal, A., Palsetia, D., et al. (2011). Ses: Sentiment elicitation system for social media data. In *Data mining workshops (ICDMW), 2011 IEEE 11th international conference on* (pp. 129–136). IEEE.

Zhang, P., & He, Z. (2015). Using data-driven feature enrichment of text representation and ensemble technique for sentence-level polarity classification. *Journal of Information Science, 41*, 531–549.