Mining the Opinionated Web: Classification and Detection of Aspect Contexts for Aspect Based Sentiment Analysis

Oscar Araque, Ganggao Zhu, Manuel García-Amado and Carlos A. Iglesias
Univeridad Politécnica de Madrid
Escuela Técnica Superior de Ingenieros de Telecomunicación
Departamento de Ingeniería Telemática
Spain, Madrid
Email: oscar.aiborra@alumnos.upm.es, gzhu@dit.upm.es,
manuel.garcia-amado.sancho@alumnos.upm.es, cif@dit.upm.es

***Abstract*—Aspect Based Sentiment Analysis (ABSA) provides further insight into the analysis of social media. Understanding user opinion about different aspects of products, services or policies can be used for improving and innovating in an effective way. Thus, it is becoming an increasingly important task in the Natural Language Processing (NLP) realm. The standard pipeline of aspect-based sentiment analysis consists of three phases: aspect category detection, Opinion Target Extraction (OTE) and sentiment polarity classification. In this article, we propose an alternative pipeline: OTE, aspect classification, aspect context detection and sentiment classification. As it can be observed, the opinionated words are first detected and then are classified into aspects. In addition, the opinionated fragment of every aspect is delimited before performing the sentiment analysis. This paper is focused on the aspect classification and aspect context detection phases and proposes a twofold contribution. First, we propose a hybrid model consisting of a word embeddings model used in conjunction with semantic similarity measures in order to develop an aspect classifier module. Second, we extend the context detection algorithm by Mukherjee et al. to improve its performance. The system has been evaluated using the SemEval2016 datasets. The evaluation shows through several experiments that the use of hybrid techniques that aggregate different sources of information improves the classification performance.**

I. Introduction

An increasing number of users utilizes web sites and social media to share their experiences and degree of satisfaction with products, services or places, among others. On-line opinionated reviews are an important source of customer feedback that companies can use in order to measure satisfaction and even improve their products and services. Also, user-generated content in web sites and social networks has experimented an important growth [1]. This has contributed largely to the development of the Sentiment Analysis (SA) field. More concretely, Aspect Based Sentiment Analysis (ABSA) is the problem of mining opinions from text about specific entities and their associated aspects [1]. ABSA techniques require a more granular vision of the opinion mining problem as not only sentiment polarity is estimated, but also requires the aspects are identified and analysed. For example, an ABSA system that is presented with the text "*The food was lousy - too sweet or too salty and the portions tiny.*" should express that both aspects, *food* and *portion*, are associated with a negative polarity.

In this article we present a complete ABSA system that addresses the different parts of the problem through a modular architecture, where each piece tackles a single task. The system consists of four phases. First of all, the *aspect detection* module is in charge of detecting the words that are referring to an opinion. That is, that words (or word) that form an aspect. Secondly, the *aspect classification* module classifies the detected aspect into one of several possible topics in the domain. Next, the *context detection* module determines the aspect context boundary. Finally, the *sentiment estimation* module realizes a sentiment analysis of the opinion and its aspect, yielding the estimated polarity.

The rest of this paper is organized as follows. In Sect. II the related work to our proposal is summarized. Sect. III presents the architecture of the system as a whole, briefly describing the aim of each module. Sect. IV describes our proposal for combining knowledge and corpus sources for aspect classification. Following, our proposal for context detection is presented in Sect. V. Sect. VI presents the sentiment estimation module. In order to evaluate the proposed system, Sect. VII depicts the experimental results obtained. Finally, conclusions and future work are presented in Sect. VIII.

II. Related Work

Many approaches aim to detect the global sentiment polarity of a document or a sentence, but efforts have been made to analyze the sentiment at the aspect level [2], [3]. In this context, our work presents a hybrid system that classifies both the aspect and its opinion. As indicated by [4], aspect based sentiment analysis has typically two steps: (i) identification and extraction of the aspects included in an opinion sentence, and (ii) estimating the sentiment polarity of said aspects. This work deals with these two problems. In this section, the related work of both problems is summarized.

A. Aspect Classification

In the context of topic classification, Latent Dirichlet Allocation (LDA) driven models can be used [4]. LDA is a generative probabilistic model that considers each document as a mixture of topics. Added to this, many variations of this topic modeling method have been presented, such as pLDA [5]. In similar lines, [6] proposes a multi-grain approach to extract opinion aspects (MG-LDA), extending the LDA approaches. The MG-LDA method extracts the opinion aspects and it also clusters them into topics.

Some additional methods for aspect classification are the semantic-based approaches. These techniques use the concept of *semantic relatedness* to aid in many Natural Language Processing tasks. Basically, the problem is to determine the relation between concepts or words. That is, it is aimed to create a way of measuring the distance between the aspect words and certain topics. In this context, some curate knowledge sources (lexical databases) can be useful, such as WordNet [7]. Also, more simple resources have been used, as the network-based dictionary approach proposed in [8].

B. Sentiment estimation

The dominant approaches on sentiment analysis are driven by *machine learning* methods [3], [9]. The most common approach consists on the Bag of Word (BOW) model, where each document is transformed into a feature vector that is then fed to a classification algorithm. Other types of features are usually used, such as Part of Speech (POS) tagging, which is a elemental model of syntactic analysis [10]. An statistical approach for representing documents is known as TF-IDF, where words are weighted depending on their frequency on the corpus [11]. Moreover, many sentiment analysis design involve the use of a *sentiment lexicon* as source of subjective information [12]. Nevertheless, lexicon-based approaches have many drawbacks: the necessity of labeled data that is reliable and consistent, the expression differences between domains and the fact that lexicons cannot be automatically translated for multilingual use [13]. Also, extracting non-simple features from text and figuring out which ones are relevant is a fundamental question in the machine learning driven techniques [14].

Alternatively, *deep learning* techniques have shown promising performance in many NLP tasks, including sentiment analysis [15]. One common use of deep learning is to learn complex features from the data with minimum external contribution through deep neural networks [16]. Continuous representations of words as vectors, also known as word embeddings have been used for sentiment analysis [17]. Besides, one interesting approach is to augment the knowledge contained in these word embeddings with other sources of information. This added information can be sentiment specific word vectors [17], or a concatenation of manually crafted features with word vectors [18]. Another approach that incorporates new information to the embeddings consists on extract sentiment features in conjunction with semantic features [19].

In addition to these approaches, ensemble methods can be used for improving sentiment classifications performance. Ensemble methods combine the predictions of various classifiers (base classifiers) and apply some function on them in order to yield a final prediction. Rule based ensemble, such as majority voting, can be quite effective in the task of sentiment classification [20]. Besides, extra subjective knowledge can be added with ensemble techniques, such as POS using a rule based ensemble model [21]. Also, a meta classifier ensemble model can be used, as in [20]. Meta learning models are based on the use of base classifiers predictions as features fed to an additional classifier that predicts the polarity.

III. System Overview

The proposed system is divided into four modules, each one addressing a different dimension of the ABSA problem: (i) the *aspect detection module* that detects OTE in an opinionated text; (ii) the *aspect classification module* that classifies each detected target into one of several possible classes in the studied domain; (iii) the *context detection module* that determines the limits of the opinion text for the detected aspect; and (iv) the *sentiment analysis module* is in charge of computing the sentiment polarity of the target based on the detected context.

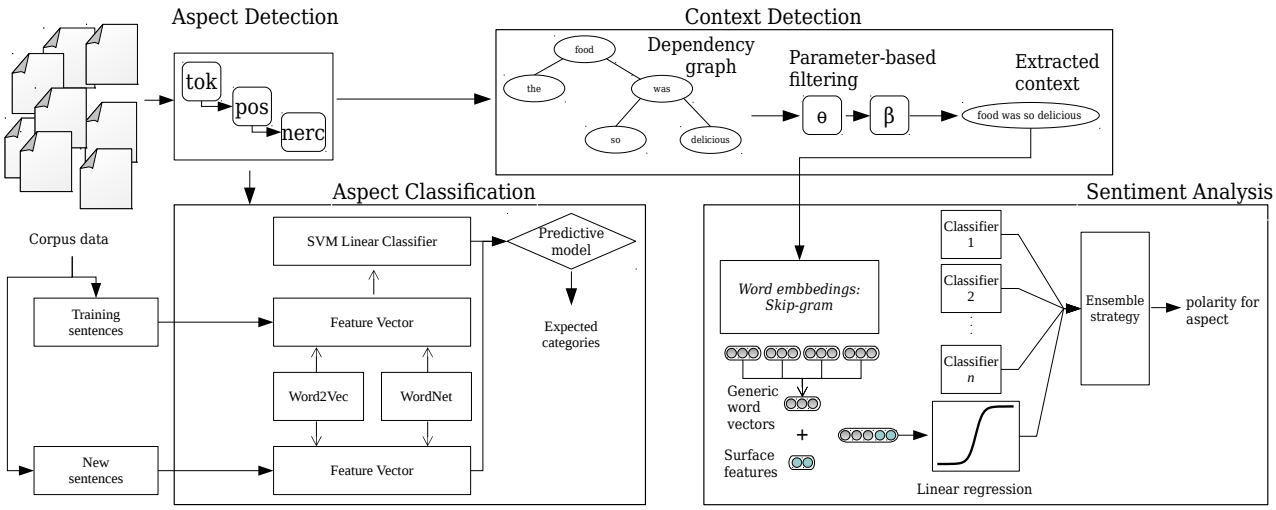
Figure 1 illustrates the data flow of the system and its main components. The process of Aspect Based Sentiment Analysis is as follows. Firstly, the text data is analyzed by the aspect detection module. This module is the one proposed in [22], composed of several sub-modules called *pipes*. Each pipe realizes a specific function and, when combined, they yield the set of words that represent an aspect in the opinion text. That is, the combination of these pipes allows us to detect the aspects. The *ixa-pipe-tok* tokenizes and segments the text, *ixa-pipe-pos* performs POS tagging and lemmanizes, and *ixa-pipe-nerc* performs the Opinion Target Extraction (OTE). We have used the already trained models offered by the tools, as they are prepared for the restaurant-reviews domain.

The, the detected aspects are passed to the aspect classification (Sect. IV) and context detection (Sect. V) modules. Finally, the context detection module requests analyses to the sentiment estimation (Sect. VI) module.

IV. Aspect Classification

Aspect Category Detection is a sub-task of ABSA, aiming to identify every entity *E* and attribute *A* pair, towards which an opinion is expressed in the given text [23]. Specifically, given an input sentence such as "The food was delicious", the aspect category detection extracts the E and A pair (e.g., category=FOOD#QUALITY) for the target word "food". We have chosen the English restaurants domain of the ABSA of SemEval2016 [23]. In the restaurant domain, SemEval predefines a set of entity labels (SERVICE, RESTAURANT, FOOD, DRINKS, AMBIANCE, LOCATION) and a set of attribute labels (GENERAL, PRICE, QUALITY, STYLE_OPTION, MISCELLANEOUS). The entities and labels compose 12 categories. Our task of aspect category classification consists in assigning a aspect category to the opinion target words.

Fig. 1. Pipeline of the proposed Aspect-based Sentiment Analysis System



The baseline of this aspect category classification provided by SemEval employs a Support Vector Machine (SVM) with a linear kernel. Specifically, n unigram features are extracted from the training data, where the category value (e.g., FOOD#QUALITY) of the tuple is used as the correct label of the feature vector [23]. For each test sentence s, a feature vector is built and the trained SVM is used to predict the correct category. This unigram feature representation lacks of the ability to address those feature words that are not encountered in the training process. As reported in SemEval [23], word clusters learned from Yelp data are used to expand the features. However, those similar words of word clusters are added to feature vectors considering the same weight as the unigram features appearing in the training data, without concerning the different semantic distance between words.

With such concerns, we aim at combining knowledge (e.g. WordNet) and corpus (e.g. Yelp) sources in order to improve aspect classification. Our main contribution is the hybrid model that consists of a word embeddings model [24] and semantic similarity model using WordNet [25]. We propose to use similarity score as the weight of each vector dimension so that the semantic similarity between words computed by word2vec and semantic similarity measures are included for training. Specifically, we explicitly use the n unigrams as feature vector, in which the word similarity between target words and feature words are used to represent each dimension of feature vector. The idea is to train a semantic predictive model for each category based on the feature words and similarity models using SVM. Formally, let $F = \{f_1, f_2, \ldots, f_n\}$ be the set of feature words, a feature vector is represented as $V \in [0,1]^N$. For a set of target words $T = \{w_1, \ldots, w_m\}$, the value of a dimension f_i is computed from $\max_{w_j \in T} sim(w_j, f_i)$, where the sim function denotes the word similarity between two words. The calculation of similarity scores is more computational intensive than counting the occurrence of words. Since the target words are in the form of short text (several words), and the feature vector can be composed by most representative words (small vector dimensions), the intensive computation problem can be alleviated using word similarity matrix.

The sim function is implemented by word2vec [24] for training Yelp data and the semantic similarity measures based on WordNet [25]. For word2vec, we have obtained a continuous representation of words, where words that co-occur frequently are mapped to vectors close in vector space. Based on the distributional semantics hypothesis, the words co-occur in a same surrounding context are treated as relevant so that they have high similarity. Consequently, the $sim(w_j, f_i)$ function is implemented as cosine similarity between two word vectors. Using this word2vec similarity model, a first feature vector $V_{word2vec} \in [0,1]^N$ is obtained.

The word2vec model considers the co-occurrence information of the same surrounding context, which would make a wide variety of words to be considered as related. This would challenge the word2vec model when discriminating words from different categories that are frequently collocated (e.g. food and drink). For instance, in restaurant domain, those target words such as fish and wine would appear in same surrounding contexts (e.g. "the fish is delicious and the wine is great"). If a word2vec model is trained from such corpus simply based on calculating co-occurrences of words, many words belonging to different categories would have similar similarity. In order to solve this problem, semantic similarity methods using WordNet [25] are useful to complement the word2vec model by including the structural knowledge from taxonomy. As illustrated in a fragment of WordNet in Fig. 2, lamb, beef, and seafood are sub-concepts of FOOD category, while coffee, tea and milk are sub-concepts of DRINKS category. Although WordNet based similarity model can retain taxonomical information from WordNet, it can only address limited words that are contained in WordNet. Combining

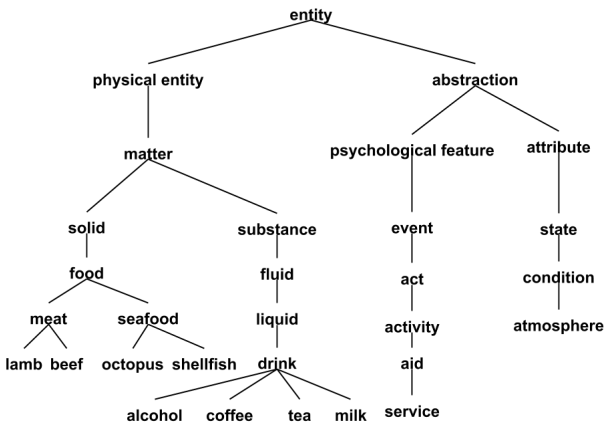


Fig. 2. A Fragment of WordNet Concept Taxonomy

word2vec and WordNet similarity models can enable the aspect classification model to have good ability in addressing large vocabularies and encoding hierarchical knowledge of common words from WordNet. In consequence, apart from Word2Vec, we also consider the semantic similarity methods using WordNet.

The semantic similarity methods exploit the hierarchical classification of all words via is-a relation, whose intuition is that two words are more similar if they are closer to each other in WordNet taxonomy. There have been many semantic similarity measures proposed in the literature [7]. To implement the WordNet based sim function, we study some of the most common ones. The simplest semantic similarity measure is counting the number of nodes or edges (shortest path) connecting two words in WordNet taxonomy. Let $path(w_i, w_j)$ be the shortest path between w_i and w_j, the Path [26] method defines semantic similarity as:

$$sim_{Path}(w_i, w_j) = \frac{1}{1 + path(w_i, w_j)} \tag{1}$$

The Leacock-Chod [27] method measures the semantic similarity between words based on their shortest path length using a non-linear function illustrated in Eq.(2):

$$sim_{Leacock-Chod}(w_i, w_j) = -\log\left(\frac{length(w_i, w_j)}{2D}\right) \tag{2}$$

where D is the maximum depth of the taxonomy. The idea of using depth information lies in the property of taxonomies that the upper-level words in a taxonomy are supposed to be more general. For example in Fig. 2, the word pair *lamb* and *beef* are more similar than the word pair *meat* and *seafood*. The Wu & Palmer method measures the depth of two words in a taxonomy with the Least Common Subsumer (LCS), which is the most specific word that is a shared ancestor of the two words. For example, the LCS of word *beef* and word *octopus* is the word *food*. Let w_{lcs} be the LCS of words w_i and w_j, then the Wu & Palmer [28] method measures semantic similarity of given words using the following formula:

$$sim_{Wu\&Palmer}(w_i, w_j) = \frac{2depth(w_{lcs})}{depth(w_i) + depth(w_j)} \tag{3}$$

The above knowledge-based semantic similarity methods consider the structure of a taxonomy which have a common drawback of uniform distance between words in the taxonomy. Some other approaches consider the Information Content (IC) to solve the uniform distance drawback. The IC of a word is given by the probability of encountering the word in a corpus. Note that we use Brown Corpus [29] for WordNet to compute IC. The Resnik [30] method relies on the IC of LCS node of two words.

$$sim_{Resnik}(c_i, c_j) = IC_{corpus}(c_{lcs}) \tag{4}$$

The consequent works by Lin [31] and Jiang & Conrad [32] extend the IC-based method by including the IC of words.

$$sim_{Lin}(w_i, w_j) = \frac{2IC(w_{lcs})}{IC(w_i) + IC(w_j)} \tag{5}$$

$$sim_{Jiang\&Conrad}(w_i, w_j) = \frac{1}{1 + IC(w_i) + IC(w_j) - 2IC(w_{lcs})} \tag{6}$$

The IC-based methods lack of important information of path and depth. In order to chose the best WordNet based semantic similarity method for the aspect category classification, we will experiment with all the semantic similarity described above in evaluation.

As illustrated in Fig. 1, a list of feature words are extracted from training data. Apart from the word2vec based feature vector $V_{word2vec}$ mentioned previously, another feature vector $V_{wordnet} \in [0, 1]^N$ is composed by computing the semantic similarity between target words and feature words using the WordNet based semantic similarity methods. Consequently, a $2N$ dimension vector is composed for training and classifying new sentences by considering both word2vec similarity model and WordNet similarity model. The evaluation of this module is presented in Sect. VII-B. The main results show that combining word embedding and semantic similarity measures can improve the performance of aspect category classification.

V. Aspect Context Detection

Aspect Context Detection is the task of detecting the text fragment in the original text that corresponds to the opinion about an attribute *A* of an entity *E*. For the aspect context detection, we have modified the algorithm proposed by Mukerjee et al. [33] in order to improve its performance. The original algorithm is based on computing the distance between words through dependency parsing. In this way, these distances can be represented in a graph, allowing the computation of the aspect context. This context detection technique is based in the assumption that the more closely associated words come together to express an opinion regarding a certain aspect. If n aspects $(a_1, a_2, ..., a_n)$ have been detected in an opinion, the

algorithm for extracting the set of words w_i that express any opinion about the target aspect a_t proceeds as described in Algorithm 1.

Algorithm 1 Dependency extraction algorithm

1) Initialize n clusters $C_i \forall i = 1..n$
2) Make each $a_i \in A$ the clusterhead of C_i. The target aspect a_t is the clusterhead of C_t. Initially, each cluster only consists of the clusterhead.
3) Assign each word w_j to cluster C_k s.t. $k = arg\ min_{i \in n} d(w_j, a_i)$
4) Merge any cluster C_i with C_t if $d(a_i, a_t) < \theta$, where θ is some threshold distance.
5) The set of words $w_i \in C_t$ expresses the opinion regarding the target aspect a_t.
6) If $\beta \neq 0$, add to (or remove from) C_t the words w_p so that

if $\beta > 0$:

$$\max d(w_i, a_t) < d(w_p, a_t)$$
$$d(w_p, a_t) \leq \max d(w_i, a_t) + \beta$$

if $\beta < 0$:

$$\max d(w_i, a_t) + \beta \leq d(w_p, a_t)$$
$$d(w_p, a_t) < \max d(w_i, a_t)$$

The original formulation of this algorithm includes a threshold parameter (θ) that controls the association of similar opinion contexts. Added to this, we have generalized this algorithm by adding one more parameter (β) that modifies the behaviour of the algorithm. This generalization intends to improve the sentiment analysis performance of the system by expanding or reducing the number of words that are included in aspect contexts. After the context is detected, this additional parameter controls the obtained context, adding or removing context words. Regarding this, we do such attending to the distances in the computed dependency graph. It consists on the value obtained after the generation of the dependency graph, the measure of the distance in it.

This β parameter controls the number of words that are either added ($\beta > 0$) or removed ($\beta < 0$) from the aspect context considering the distances on the dependency graph. Algorithm 1 shows the modified method with this type of distance. The associated parameter differs from the original formulation of the algorithm [33] when its value is not zero. That is, when $\beta = 0$, our proposal is identical to that of the original.

VI. Aspect Based Sentiment Analysis

In this module, the detected context (Sec. V) is used for the sentiment estimation of the aspect. That is, we consider that the text that is contained in the detected context refers to the analyzed aspect, and thus that is what the sentiment estimation modules uses.

The sentiment classification has been addressed with the use of previously proposed models. These sentiment analysis models aim to take advantage of different types of features, assuming that a sentiment classifier can yield a better performance in the sentiment analysis task when it is provided with a higher quantity and variety of information. For this end, two different combination techniques are used: ensemble of classifiers and ensemble of features.

On one hand, the ensemble of classifiers combines the predictions of the classifiers that form the ensemble (base classifiers). In this way, the augmented information is given through each classifier's sentiment prediction. On the other hand, the ensemble of features combines the word vectors or features that have been extracted differently. With this, different sources of information are inserted into a single classifier.

The features used in this work are: generic word vectors, representations obtained through a word embeddings algorithm; and surface features, such as sentiment lexicons and Part-of-Speech tagging. The models we use in this work are described next.

Generic word vectors model (M_G). This model combines the vectors from each word of the analyzed document, and aggregates them into a single vector. The aggregation functions that are used are the *average*, *max* and *min*. The generic word vectors are obtained using the *skip-gram* model [24]. Once the aggregated vectors have been composed, they are fed to a linear regression algorithm, that yields the sentiment polarity. Unlike the following models, M_G does not combine different sources of information.

Ensemble of classifiers (CEM_{SG}) model. This model ensembles the predictions of a number of classifiers that have been trained with both surface features and generic word vectors, as in the M_G model. The ensemble techniques used are two. First, a fixed rule strategy known as majority voting, where the polarity class is decided by the voting results of the base classifier. In case of tie, the positive polarity is selected. Second, a meta-learning technique, where the predictions of the base classifiers are used as features for a meta learner that yields the final sentiment prediction. In this work, the meta classifier is implemented using the Random Forest algorithm. The base classifiers of the ensemble are the same as in [34].

Ensemble of features (M_{SG}), en ensemble of features model. In this model, the attempt to improve the M_G proposal is tackled by combining the previously used generic word vectors and a set of surface features. Both types of features are combined by concatenation, obtaining a enlarged vector. This vector, as in M_G model, is then fed to a linear regression classifier, that predicts the sentiment polarity. The surface features used in this work are: Wordnet-Affect lexicon values [35], number of exclamation and interrogation marks, number of positive, neutral and negative words, number of words that are in caps and number of elongated words.

VII. Evaluation

In order to evaluate the aspect classification, context detection and sentiment analysis sub-modules, we have performed several experiments. In these experiments, we aim to evaluate

TABLE I
STATISTICS OF THE USED DATASETS.

Dataset	#Positive	#Negative
Yelp-extracted	1,492,558	450,540
SemEval16 train	1,696	773
SemEval16 test	609	204

the effectiveness of the proposed system and, also, optimize some of the described parameters. The metric used is the F-score.

A. Datasets

For this evaluation, we have extracted a dataset that is aligned with the restaurant reviews domain from the Yelp Challenge dataset [1]. This dataset provides with a high quantity of data that can be used for the word embeddings training. Also, we labeled this dataset using a distant supervision strategy for the sentiment polarity. That is, we have taken advantage of the Yelp start-based rating, considering 1 or 2 stars as negative sentiment, and 4 or 5 start as positive polarity. In this work, we do not consider the role of the neutral polarity.

Also, we have used as development dataset (learning of hyper-parameters) the SemEval16 training data, and as test dataset (final validation of the sentiment performance) the SemEval16 test set. These two datasets, as well as the one extracted from Yelp, are summarized in Table I.

B. Aspect Category Classification Evaluation

We use the SemEval16 dataset of English Restaurant domain dataset. The training dataset consists of 1880 tuples and the test dataset consists of 650 tuples. We extracted most common 10 words of each category and composed into 76 feature words by removing duplicates. The small feature number is not a problem since the vocabularies are contained in word2ve and WordNet. Nevertheless, the quality of feature words should be considered because we use the word similarity scores as the value of feature vectors. We use the most frequent words for simplicity in this article. The word2vec similarity model andWordNet similarity model are used to compute word similarity between target words and feature words. We trained the aspect classification model using the linear kernel of SVM using the sklearn[2] package. The classification metrics accuracy, precision, recall and F-score are used as the performance metrics to evaluate the different models.

We have experimented with the classification model in different settings: simple feature, knowledge-based feature, dense vector feature and combined features. The experimental results are shown in Table II. In the simple feature, we use the simple word list features $V_{wordlist} \in \{0,1\}^N$, where the word list is the 76 feature words. In this setting, we use the unigram occurrence feature to train a classification module using SVM, and use this model as baseline. Note that the different learning softwares and settings would inflence the experimental results so that we implemented a simple baseline following the description of SemEval. In order to show that the similarity based feature is more effective than the simple word occurrence feature, we extended the simple feature model to the knowledge-based model and dense vector model. In the knowledge-based setting, we have trained and evaluated the classification model using the WordNet based similarity measures respectively. Table II shows that the Path [26] similarity measure is the best metric for aspect classification, and the most of similarity measures are more effective than the baseline except for the Resnik [30] method. In the dense vector setting, we have used word2vec embedding to learn the word vectors from Yelp comments data, and trained the aspect classification model only with the word2vec similarity model. The experimental result shows that the word2vec similarity model is more effective than knowledge-based methods and baseline. By looking at each category, we found that the knowledge-based features are more effective for food and drink categories while word2vec performs better in other categories. Since word2vec feature is trained from a domain corpus (Yelp comments), it has better coverage in vocabularies and the categories such as AMBIENCE, LOCATION are more concerned with relevant features rather than hierarchical feature. In the combined setting, we use both word2vec similarity model and WordNet similarity model to train and evaluate in order to select the best combination between word embedding and semantic similarity methods. Table II shows that both Path [26] and Jiang & Conrad [32] are the best in combining with word2vec, in terms of F measure (.79).

TABLE II
ACCURACY, PRECISION, RECALL AND F-MEASURE OF ASPECT CATEGORY CLASSIFICATION USING DIFFERENT METHODS.

Method	Corpus & KB	Accuracy	Precision	Recall	F-measure
Simple Feature	Word List	.745	.72	.74	.71
Knowledge-based					
Path. [26]	WordNet	.78	.77	.78	.75
Leacock-Chod. [27]	WordNet	.757	.73	.76	.73
Wu & Palmer. [28]	WordNet	.751	.70	.75	72
Resnik. [30]	WordNet	.646	.65	.65	.63
Lin. [31]	WordNet	.774	.73	.77	.74
Jiang & Conrad. [32]	WordNet	.768	.77	.77	.74
Dense Vectors					
Word2Vec. [24]	Yelp	.818	.79	.82	.78
Combination					
Word2Vec + Path	WordNet + Yelp	.82	.80	.82	.79
Word2Vec + Leacock-Chod	WordNet + Yelp	.81	.80	.81	.78
Word2Vec + Wu & Palmer	WordNet + Yelp	.813	.80	.81	.78
Word2Vec + Resnik	WordNet + Yelp	.814	.80	.81	.78
Word2Vec + Lin	WordNet + Yelp	.813	.80	.81	.78
Word2Vec + Jiang & Conrad.	WordNet + Yelp	.82	.80	.82	.79

In summary, from the experimental results, we found that the similarity based features are effective in learning the aspect classification model. Furthermore, combining the word embedding model and semantic similarity measure is promising in training aspect classification model, since it has achieved best performance in our experiments, and it can combine the word coocurrence information together with hiearchical knowledge from WordNet.

[1]https://www.yelp.com/dataset_challenge

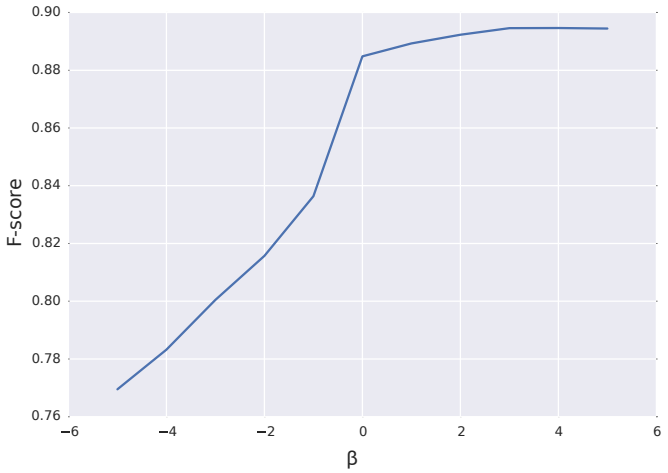
[2]http://scikit-learn.org

C. Context detection validation

In these context detection experiments, we have taken as performance metric the sentiment F-score on the development set with the M_G sentiment classification model. In this way, the different contexts that vary from the variation of several parameters are fed to this sentiment model. As for the training of the sentiment classifier, it is explained further in Sect. VII-D.

The parameter validated in the context detector module is the β value, that controls the words that added or removed to the context. As can be seen in Figure 3, the sentiment performance increases when $\beta > 0$. The improvement between $\beta = 0$ (original formulation) and $\beta = 6$ (maximum improvement of our proposal) is of 3.04% of the F-score.

Fig. 3. Variation of the F-score with the β parameter of the context detector.



D. Sentiment training and validation

Firstly, the *skip-gram* model has been trained with the Yelp-extracted dataset, setting the dimension of the resulting vectors to 400, and a minimum count of 5. As this is unsupervised training, polarity labels have not been used.

After the training of the word embeddings model, the M_G model requires that a linear regression algorithm is trained. For this, we used the aggregated vectors for each document of the Yelp-extracted dataset, and the corresponding distant sentiment labels. Also, the set of possible aggregation functions on the M_G model has been evaluated on the SemEval16 train data, obtaining the best performance with the *average* function a 91.79 % of F-score. The next high performance in this sense is the combination *max+avg*, with a 87.94 %.

The ensemble of classifiers model does not need a training process, as it is composed of already trained classifiers. Nevertheless, the meta-learning strategy does need training, as it learns from the predictions of its base classifiers. For this end, the meta classifier is trained with the development data.

Finally, Table III shows the F-scores for the different sentiment models in the SemEval16 test dataset. CEM^{Vo}_{SG} is the ensemble of classifier with the majority voting scheme, and CEM^{MeL}_{SG} with the meta learning strategy. The BOW baselines are tested and compared with the proposed models. It can be seen that the TF-IDF does not improve the sentiment performance in these experiments. Also, the best performing model is the M_{SG} model. The experiment result indicates that joining generic word vectors and surface features through a feature ensemble strategy improves the sentiment performance. Nevertheless, the two classifier ensemble methods do not result in a classification improvement, but a performance decrease.

TABLE III
FSCORES OBTAINED IN THE SEMEVAL16 TEST DATASET.

Model	**F-Score** (binary)
BOW + TF-IDF + SVM	81.29
BOW + SVM	81.78
M_G (*skipgram*)	90.76
M_G + TF-IDF	90.67
CEM^{Vo}_{SG} (*skipgram* + voting)	87.54
CEM^{MeL}_{SG} (*skipgram* + meta-learning)	81.75
M_{SG} (*skipgram* + surface features)	**91.00**

To the extent of our knowledge, there is no public disaggregation of F-score for the predicted classes on the three best systems. Nevertheless, we can compare to the proposal described in [36], that claim their F-score for positive and negative classes are 88.26 and 76.21%, respectively. Our systems performs better in relation to the positive class, with a 91.00%, but it does not perform better on the negative class, yielding 73.91%. For both systems, the F-score metric for the neutral class is 0.

VIII. CONCLUSIONS AND FUTURE WORK

This paper presents an Aspect Based Sentiment Analysis system divided into four modules, each one addressing a single step of the ABSA problem: aspect detection, aspect classification, aspect context detection and sentiment estimation.

For the aspect classification module we proposed a hybrid approach where both word embeddings and semantic similarity measures are used. The experiments show that the combination of these two types of features improves the classification when compared to these same techniques separately.

The aspect context detection module uses a modified dependency parsing tree algorithm whose assumption is that close words in the dependency tree express an opinion of the same aspect. A generalization parameter is introduced into the algorithm and evaluated on the data, finding that this addition on the detected context methods improves the sentiment classification performance.

The sentiment estimation module is composed by a hybrid system that uses a configurable combination of word embeddings, traditional sentiment features and an ensemble of classifiers. The combination of traditional sentiment features (e.g., sentiment lexicon values) and *skipgram* word embeddings is shown to improve the sentiment performance of the system.

All in all, we have addressed in this paper how the combination of knowledge and corpus sources can improve both aspect classification and polarity detection, being complemented by similarity metrics in aspect classification. Moreover, we have explored how modifying the scope of the aspect context

affects context detection, and have proposed a generalisation of Mukherjeer et al. algorithm that can be used for its optimisation in other datasets.

As future work, we consider that the aspect classification module performance can be increased if using a domain adapted semantic knowledge base. In this way, the studied domain can be better characterized. Regarding the sentiment estimation, a neural approach can be taken to train specific word vectors adapted to the domain, instead of using generic word vectors. Also, we have intentions of expanding these systems to other domains, such as emotion analysis.

ACKNOWLEDGEMENT

This research work is supported by the EC through the H2020 project MixedEmotions (Grant Agreement no: 141111) and by the Spanish Ministry of Economy under the R&D project Semola (TEC2015-68284-R).

REFERENCES

[1] B. Liu, *Sentiment analysis: Mining opinions, sentiments, and emotions*. Cambridge University Press, 2015.

[2] S. Poria, E. Cambria, and A. Gelbukh, "Aspect extraction for opinion mining with a deep convolutional neural network," *Knowledge-Based Systems*, vol. 108, pp. 42–49, 2016.

[3] S. Poria, E. Cambria, L.-W. Ku, C. Gui, and A. Gelbukh, "A rule-based approach to aspect extraction from product reviews," in *Proceedings of the second workshop on natural language processing for social media (SocialNLP)*, 2014, pp. 28–37.

[4] M. Cataldi, A. Ballatore, I. Tiddi, and M.-A. Aufaure, "Good location, terrible food: detecting feature sentiment in user-generated reviews," *Social Network Analysis and Mining*, vol. 3, no. 4, pp. 1149–1163, 2013.

[5] Y. Wang, H. Bai, M. Stanton, W.-Y. Chen, and E. Y. Chang, "Plda: Parallel latent dirichlet allocation for large-scale applications," in *International Conference on Algorithmic Applications in Management*. Springer, 2009, pp. 301–314.

[6] I. Titov and R. McDonald, "Modeling online reviews with multi-grain topic models," in *Proceedings of the 17th international conference on World Wide Web*. ACM, 2008, pp. 111–120.

[7] A. Budanitsky and G. Hirst, "Evaluating wordnet-based measures of lexical semantic relatedness," *Computational Linguistics*, vol. 32, no. 1, pp. 13–47, 2006.

[8] H. Kozima and T. Furugori, "Similarity between words computed by spreading activation on an english dictionary," in *Proceedings of the sixth conference on European chapter of the Association for Computational Linguistics*, 1993, pp. 232–239.

[9] S. Wang and C. D. Manning, "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*, 2012, pp. 90–94.

[10] K. Gimpel, N. Schneider, B. O'Connor, D. Das, D. Mills, J. Eisenstein, M. Heilman, D. Yogatama, J. Flanigan, and N. A. Smith, "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*, ser. HLT '11. Stroudsburg, PA, USA: Association for Computational Linguistics, 2011, pp. 42–47.

[11] J. Martineau and T. Finin, "Delta tfidf: An improved feature space for sentiment analysis." *ICWSM*, vol. 9, p. 106, 2009.

[12] P. Melville, W. Gryc, and R. D. Lawrence, "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*, ser. KDD '09. New York, NY, USA: ACM, 2009, pp. 1275–1284.

[13] M. Taboada, J. Brooke, M. Tofiloski, K. Voll, and M. Stede, "Lexicon-based methods for sentiment analysis," *Computational linguistics*, vol. 37, no. 2, pp. 267–307, 2011.

[14] A. Agarwal, B. Xie, I. Vovsha, O. Rambow, and R. Passonneau, "Sentiment analysis of twitter data," in *Proceedings of the workshop on languages in social media*. Association for Computational Linguistics, 2011, pp. 30–38.

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

[16] Y. Bengio, "Learning deep architectures for ai," *Foundations and trends® in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

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

[18] D. Tang, F. Wei, B. Qin, T. Liu, and M. Zhou, "Coooolll: A deep learning system for twitter sentiment classification," in *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, 2014, pp. 208–212.

[19] Z. Su, H. Xu, D. Zhang, and Y. Xu, "Chinese sentiment classification using a neural network tool word2vec," in *Multisensor Fusion and Information Integration for Intelligent Systems (MFI), 2014 International Conference on*, Sept 2014, pp. 1–6.

[20] R. Xia, C. Zong, and S. Li, "Ensemble of feature sets and classification algorithms for sentiment classification," *Information Sciences*, vol. 181, no. 6, pp. 1138 – 1152, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0020025510005682

[21] R. Xia and C. Zong, "A pos-based ensemble model for cross-domain sentiment classification." in *IJCNLP*. Citeseer, 2011, pp. 614–622.

[22] R. Agerri, J. Bermudez, and G. Rigau, "Ixa pipeline: Efficient and ready to use multilingual nlp tools." in *LREC*, vol. 2014, 2014, pp. 3823–3828.

[23] M. Pontiki, D. Galanis, J. Pavlopoulos, H. Papageorgiou, I. Androutsopoulos, and S. Manandhar, "Semeval-2016 task 5: Aspect based sentiment analysis," in *Proceedings of the 8th international workshop on semantic evaluation (SemEval 2014)*. Citeseer, 2016, pp. 27–35.

[24] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[25] R. Mihalcea, C. Corley, and C. Strapparava, "Corpus-based and knowledge-based measures of text semantic similarity," in *AAAI*, vol. 6, 2006, pp. 775–780.

[26] R. Rada, H. Mili, E. Bicknell, and M. Blettner, "Development and application of a metric on semantic nets," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 19, no. 1, pp. 17–30, 1989.

[27] C. Leacock and M. Chodorow, "Combining local context and wordnet similarity for word sense identification," *WordNet: An electronic lexical database*, vol. 49, no. 2, pp. 265–283, 1998.

[28] Z. Wu and M. Palmer, "Verbs semantics and lexical selection," in *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, ser. ACL '94. Stroudsburg, PA, USA: Association for Computational Linguistics, 1994, pp. 133–138.

[29] W. N. Francis and H. Kucera, "Brown corpus manual," *Brown University*, 1979.

[30] P. Resnik, "Using information content to evaluate semantic similarity in a taxonomy," in *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 1*, ser. IJCAI'95. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1995, pp. 448–453.

[31] D. Lin, "An information-theoretic definition of similarity," in *Proceedings of the Fifteenth International Conference on Machine Learning*, ser. ICML '98. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1998, pp. 296–304.

[32] J. J. Jiang and D. W. Conrath, "Semantic similarity based on corpus statistics and lexical taxonomy," *Computational Linguistics*, vol. cmp-lg/970, no. Rocling X, p. 15, 1997.

[33] S. Mukherjee and P. Bhattacharyya, "Feature specific sentiment analysis for product reviews," in *International Conference on Intelligent Text Processing and Computational Linguistics*. Springer, 2012, pp. 475–487.

[34] O. Araque, "Prototype of a Sentiment Analysis System Based on Ensemble Algorithms for Combining Deep and Surface Machine Learning Techniques," Master's thesis, ETSI Telecomunicación, June 2016.

[35] C. Strapparava, A. Valitutti *et al.*, "WordNet Affect: an affective extension of wordnet." in *LREC*, vol. 4, 2004, pp. 1083–1086.

[36] M. Chernyshevich, "Ihs-rd-belarus at semeval-2016 task 5: Detecting sentiment polarity using the heatmap of sentence," *Proceedings of SemEval*, pp. 296–300, 2016.