A Comparison of Domain-based Word Polarity Estimation using different Word Embeddings

Aitor García-Pablos, Montse Cuadros, German Rigau
Vicomtech-IK4, Vicomtech-IK4, IXA research group
agarciap@vicomtech.org, mcuadros@vicomtech.org, german.rigau@ehu.eus

Abstract

A key point in Sentiment Analysis is to determine the polarity of the sentiment implied by a certain word or expression. In basic Sentiment Analysis systems this sentiment polarity of the words is accounted and weighted in different ways to provide a degree of positivity/negativity. Currently words are also modelled as continuous dense vectors, known as word embeddings, which seem to encode interesting semantic knowledge. With regard to Sentiment Analysis, word embeddings are used as features to more complex supervised classification systems to obtain sentiment classifiers. In this paper we compare a set of existing sentiment lexicons and sentiment lexicon generation techniques. We also show a simple but effective technique to calculate a word polarity value for each word in a domain using existing continuous word embeddings generation methods. Further, we also show that word embeddings calculated on in-domain corpus capture the polarity better than the ones calculated on general-domain corpus.

Keywords: sentiment lexicon, sentiment analysis, word embedding

1. Introduction

A key point in Sentiment Analysis is to determine the polarity of the sentiment implied by a certain word or expression (Taboada et al., 2011). In basic Sentiment Analysis systems this sentiment polarity of the words is accounted and weighted in different ways to provide a degree of positivity/negativity of, for example, a customer review. In more sophisticated systems, word polarity is employed as an additional feature for machine learning algorithms. This polarity value can be a categorical value (e.g. positive/neutral/negative) or a real value within a range (e.g. from -1.0 to +1.0), and can be plugged in supervised classification algorithms together with other lexical and semantic features to help discriminating the overall polarity of an expression or a sentence. Currently words are also modelled as continuous dense vectors, known as word embeddings, which seem to encode interesting semantic knowledge. The word vectors are usually computed using very big corpora of texts, like the English Wikipedia. One of the best known systems to obtain a dense continuous representation of words is Word2Vec (Mikolov et al., 2013c). But Word2Vec is not the only one, and in fact there are already a lot of variants and many researchers working on different kinds of word embeddings (Le and Mikolov, 2014; Iacobacci et al., 2015; Ji et al., 2015; Hill et al., 2014; Schwartz et al., 2014). With regard to Sentiment Analysis, word embeddings are used as features to more complex supervised classification systems to obtain very precise sentiment classifiers (Tang et al., 2014a; Socher et al., 2013). In this paper we compare a set of existing static sentiment lexicons and dynamic sentiment lexicon generation techniques. We also show a simple but competitive technique to calculate a word polarity value for each word in a domain using continuous word embeddings. Our objective is to see if word embeddings calculated on an in-domain corpus can be directly used to obtain a polarity measure of the domain vocabulary with no additional supervision. Further, we want to see to which extent word embeddings calculated on in-domain corpus improve the ones calculated on general-domain corpus and analyse pros and cons of each compared method. The paper is structured as follows. Section 2. reviews several works related to the generation of sentiment lexicons, providing the context for the rest of the paper. Section 3. describes the lexicons and methods that will be used to make the comparison, focusing on the ones using continuous word representations. Section 4. presents the datasets used to generate some of the lexicons. Section 5. describes the experiments to compare the different approaches and discusses them. Finally the last section shows the conclusions.

2. Related Work

Sentiment analysis refers to the use of NLP techniques to identify and extract subjective information in digital texts like customer reviews about products or services. Due to the grown of the social media, and specialized websites that allow users posting comments and opinions, Sentiment Analysis has been a very prolific research area during the last decade (Pang and Lee, 2008; Zhang and Liu, 2014).

A key point in Sentiment Analysis is to determine the polarity of the sentiment implied by a certain word or expression (Taboada et al., 2011). Usually this polarity is also known as Semantic Orientation (SO). SO indicates whether a word or an expression states a positive or a negative sentiment, and can be a continuous value in a range from very positive to very negative, or a categorical value (like the common 5-star rating used to rate products). Further, the SO of a word is a useful feature to be used within a more complex Sentiment Analysis system like machine learning algorithms (Lin et al., 2009; Jaggi et al., 2014; Tang et al., 2014a).

A collection of words and their respective SO is known as sentiment lexicon. Sentiment lexicons can be constructed manually, by human experts that estimate the corresponding SO value to each word of interest. Obviously, this approach is usually too time consuming for obtaining a good coverage and difficult to maintain when the vocabulary evolves or a new language or domain must be analyzed.

Therefore it is necessary to devise a method to automate the process as much as possible.

Some systems employ existing lexical resources like WordNet (Fellbaum, 1998) to bootstrap a list of positive and negative words via different methods. In (Esuli and Sebastiani, 2006) the authors employ the glosses that accompany each WordNet synset[1] to perform a semi-supervised synset classification. The result consists of three scores per synset: positivity, negativity and objectivity. In (Baccianella et al., 2010) version 3.0 of SentiWordNet is introduced with improvements like a random walk approach in the WordNet graph to calculate the SO of the synsets. In (Agerri and Garcia, 2009) another system is introduced, Q-WordNet, which expands the polarities of the WordNet synsets using lexical relations like synonymy. In (Guerini et al., 2013) the authors propose and compare different approaches based SentiWordNet to improve the polarity determination of the synsets.

Other authors try different bootstrapping approaches and evaluate them on WordNet of different languages (Maks et al., 2014; Vicente et al., 2014). A problem with the approaches based on resources like WordNet is that they rely on the availability and quality of those resources for a new languages. Being a general resource, WordNet also fails to capture domain dependent semantic orientations. Likewise other approaches using common dictionaries do not take into account the shifts between domains (Ramos and Marques, 2005).

Other methods calculate the SO of the words directly from text. In (Hatzivassiloglou and McKeown, 1997) the authors model the corpus as a graph of adjectives joined by conjunctions. Then, they generate partitions on the graph based on some intuitions like that two adjectives joined by "and" will tend to share the same orientation while two adjectives joined by "but" will have opposite orientations.

On the other hand, in (Turney, 2002) the SO is obtained calculating the Pointwise Mutual Information (PMI) between each word and a very positive word (like "excellent") and a very negative word (like "poor") in a corpus. The result is a continuous numeric value between -1 and +1.

These ideas of bootstrapping SO from a corpus have been further explored and sophisticated in more recent works (Popescu and Etzioni, 2005; Brody and Elhadad, 2010; Qiu et al., 2011)

2.1. Continuous word representations

Continuous word representations (also vector representations or word embeddings) represent each word by a n-dimensional vector. Usually, these vector encapsulates some semantic information derived from the corpus used and the process applied to derive the vector. One of the best known techniques for deriving vector representations of words and documents are Latent Semantic Indexing (Dumais et al., 1995) and Latent Semantic Analysis (Dumais, 2004).

Currently it is becoming very common in the literature to employ Neural Networks and the so-called Deep Learning to compute word embeddings (Bengio et al., 2003; Turian et al., 2010; Huang et al., 2012; Mikolov et al., 2013c). Word embeddings show interesting semantic properties to find related concepts, word analogies, or to use them as features to conventional machine learning algorithms (Socher et al., 2013; Tang et al., 2014b; Pavlopoulos and Androutsopoulos, 2014). Word embeddings are also explored in tasks such as deriving adjectival scales (Kim, 2013).

3. Lexicons and methods

Our aim is to compare different existing sentiment lexicons and methods to find out if continuous word embeddings can be used to easily compute accurate sentiment polarity over the words of a domain, and under which conditions. The experiments are carried on two specific domains, in particular restaurants and laptops reviews.

3.1. General lexicons

The General Inquirer (GI) (Stone et al., 1966) is a very well-known manually crafted lexicon that includes the polarity of many English words. GI contains about 2000 positive and negative words. It has been used in many different research works over the past years.

On the other hand we have also used the Bing Liu's sentiment lexicon (Hu and Liu, 2004). According to the web page [2] it has been compiled and incremented over many years. It contains around 6800 words with an assigned categorical polarity (positive or negative).

3.2. Wordnet based lexicons

SentiWordnet assigns scores to each WordNet synset[3] (Esuli and Sebastiani, 2006). SentiWordNet polarity consists of three scores per synset: positivity, negativity and objectivity. In (Baccianella et al., 2010) version 3.0 of SentiWordNet is introduced with improvements like a random walk approach in the graph of WordNet. We have also used the Q-WordNet as Personalized PageRanking Vector (QWN-PPV) which propagates and ranks polarity values on the WordNet graph starting from few seed words (Vicente et al., 2014).

3.3. PMI based lexicons

Following the work at (Turney, 2002), we also have derived some polarity lexicons from a domain corpus using Pointwise Mutual Information (PMI). In few words, PMI is used as a measure of relatedness between two events, in this case the co-occurrence of words with known positive contexts. In the original Turney's work the value of co-occurrence was measured counting hits in a web search (the extinct Altavista) between words and the seed word "excellent" (for positives) and the seed word "poor".

$$\mathrm{SO}(w) = \mathrm{PMI}(w, POS) - \mathrm{PMI}(w, NEG) \quad (1)$$

$$\mathrm{PMI}(w1, w2) = \log \frac{p(w1, w2)}{p(w1) \times p(w2)} \quad (2)$$

[1] A WordNet synset in a set of synonym words that denote the same concept

[2] https://www.cs.uic.edu/\~liub/FBS/sentiment-analysis.html\#lexicon

[3] A WordNet synset in a set of synonym words that denote the same concept

Firstly, we have borrowed the lexicon generated in (Kiritchenko et al., 2014) (named NRC_CANADA in the experiment tables), which was generated computing the PMI between each word and positive reviews(4 or 5 stars in a 5-star rating) and negative reviews (1 or 2 stars), for both restaurants and laptops review datasets. Because it uses the user ratings, this approach is supervised.

As a counterpart we have calculated another PMI based lexicon, in which we employ the co-occurrence of words within a five word window with the word *excellent* (analogously with the word *terrible* for negative) to calculate the PMI score. This is potentially less accurate but requires no supervised information apart from the two seed words.

3.4. Word embedding lexicons

We have applied the popular Word2Vec (Mikolov et al., 2013a) and the Stanford Glove system (Pennington and Manning,) to calculate word embeddings. We have computed three models for each system: one in a restaurant reviews dataset, another in a laptop reviews dataset and a third one in a much bigger general domain dataset (consisting on the first billion characters from the English Wikipedia[4]).

Notice that the employed general domain dataset is pretty much bigger than the domain-based datasets. General domain dataset is a 700MB raw text file after cleaning it, while restaurants and laptop dataset only weight 28 and 40 MB respectively. General domain datasets, like the whole Wikipedia data or News dataset from online newspapers, capture very well general syntactic and semantic regularities. However, to capture in-domain word polarities smaller domain focused dataset might work better (García-Pablos et al., 2015). Also notice that at the time of writing this paper, there are appearing a lot of different techniques to calculate word embeddings that could work better than plain Word2Vec(Li and Jurafsky, 2015; Rothe et al., 2016), but due to their recent apparition are not employed in these experiments.

Restaurant dataset computed similarities		
excellent	*horrible*	*slow*
outstanding	terrible	spotty
fantastic	awful	inattentive
amazing	sucked	uncaring
exceptional	horrid	painfully
awesome	poor	neglectful
top notch	sucks	lax
great	atrocious	slower
superb	lousy	inconsistent
incredible	horrific	uneven
wonderful	yuck	iffy

Table 1: Most similar words in the word embedding space computed on restaurants reviews dataset, according to the cosine similarity, for words *excellent*, *horrible* and *slow*

[4]Obtained from `http://mattmahoney.net/dc/enwik9.zip`

Laptops dataset computed similarities		
excellent	*horrible*	*slow*
outstanding	terrible	counterintuitive
exceptional	deplorable	painfully
awesome	awful	unstable
incredible	abysmal	sluggish
excelent	poor	choppy
amazing	horrid	fast
excellant	lousy	buggy
fantastic	whining	slows
terrific	horrendous	frustratingly
superb	unprofessional	flaky

Table 2: Most similar words in the word embedding space computed on laptops reviews dataset, according to the cosine similarity, for words *excellent*, *horrible* and *slow*

In table 3.4. and table 3.4. it can be observed how the word embedding computed for restaurant and laptop domain seem to capture polarity quite accurately just by using word similarity. This is because the employed datasets are customer reviews of each domain, and the kind of content present in customer reviews helps modelling the meaning and polarity of the words (adjectives in this case). Tables show top similarities according to the cosine distance between word vectors computed by each model. Words like *excellent* and *horrible* are domain independent, and the most similar words are quite equivalent for both domains. But for the third word, *slow*, the differences between both domains are more evident. The word *slow* in the context of restaurants is usually employed to describe the service quality (when judging waiters and waitresses serving speed and skills), while in the context of laptops it refers to the performance of hardware and/or software. Another advantage versus a general domain computed model is that domain-based models will contain any domain jargon words or even commonly misspelled words (as long as it appears usually enough in the corresponding dataset). A general domain dataset is less likely to cover all the vocabulary present for any possible domain.

We have used a simple formula to assign a polarity to the words in the vocabulary, using a single positive seed word and a single negative seed word.

$$\text{pol}(w) = \text{sim}(w, POS) - \text{sim}(w, NEG) \qquad (3)$$

In the equation POS is the seed positive word for the domain represented by its corresponding word vector, and analogously NEG is the vector representation of seed negative word. In the experiments we have used domain independent seed words with a very clear and context- and domain-independent polarity, in particular *excellent* and *horrible* as positive and negative seeds respectively. sim stands for the cosine distance between word vectors. Note that this simple formula provides a real number, that in a sense gives a continuous value for the polarity. The fact of obtaining a continuous value for the polarity could be an interesting property to measure the strength of the sentiment, but for now we simply convert the polarity value to a binary

label: positive if the value is greater or equal to zero, and negative otherwise. This makes the comparison with the other examined lexicons easier.

4. Domain corpora

In order to generate the lexicons with the methods that require an in-domain corpus (i.e. the PMI based one, the Word2Vec and the GloVe) we have used corpus from two different domains.

The first corpus consists of customer reviews about restaurants. It is a 100k review subset about restaurants obtained from the Yelp dataset[5] (henceforth Yelp-restaurants). We also have used a second corpus of customer reviews about laptops. This corpus contains a subset of about 100k reviews from the Amazon electronic device review dataset from the Stanford Network Analysis Project (SNAP)[6] after selecting reviews that contain the word "laptop" (henceforth Amazon-laptops).

The corpora have been processed removing all non-content words (i.e. everything except adjectives, adverbs, verbs and nouns is removed), and words have been lowercased. For other tasks like word-analogy discovery (Mikolov et al., 2013d) or marchine translation (Mikolov et al., 2013b) every word (even those that are usually considered stopwords), but as our in-domain datasets are of reduced size[7] we remove the words that are less informative to model the polarity, like pronouns, articles or prepositions. After that, both corpora have been used to feed the target methods, obtaining their respective domain-aware results. In the case of the PMI based lexicon a score, and in the case of Word2Vec and GloVe, a vector representation of the words for each domain. In the case of Word2Vec we have employed the implementation contained in the Apache Spark Mllib library[8]. This Word2Vec implementation is based on the Word2Vec Skip-gram architecture, and we have let the default hyper-parameters and configuration[9].

5. Experiments

We have performed two different evaluations. On the one hand, we have used the domain corpora (Yelp-restaurants and Amazon-laptops) to automatically obtain a list of domain adjectives ranked by frequency. From that list we have manually selected the first 200 adjectives with context-independent positive or negative polarity for each domain[10]. Then we have manually assigned a polarity label (positive or negative) to each of the selected adjectives. From now on we will refer to these annotated adjectives restaurant-adjectives-test-set and laptop-adjectives-test-set respectively. The restaurant-adjectives-test-set contains 119 positive adjectives and 81 negatives adjectives, while laptops-adjectives-test-set contains 127 positives and 73 negatives[11].

On the other hand, we have used the SemEval 2015 task 12 datasets[12]. The first dataset contains 254 annotated reviews about restaurants (a total of 1,315 sentences). The second dataset contains 277 annotated reviews about laptops (a total of 1,739 sentences).

5.1. Manual gold-lexicon based experiments

RESTAURANTS 200 ADJ GOLD LEXICON				
Name		Posit.	Neg.	Overall
General Inquirer	Acc.	**0.935**	0.944	0.939
	Cover.	0.521	0.444	0.490
BingLiu	Acc.	**0.935**	**0.979**	**0.952**
	Cover.	0.647	0.580	0.620
SentiWordNet	Acc.	0.725	0.746	0.733
	Cover.	0.857	0.778	0.825
QWN-PPV	Acc.	0.821	0.609	0.746
	Cover.	0.706	0.568	0.650
NRC_CANADA	Acc.	0.933	0.753	0.860
	Cover.	**1.000**	**1.000**	**1.000**
PMI_WINDOW_5	Acc.	0.917	0.655	0.821
	Cover.	0.807	0.679	0.755
W2V_DOMAIN	Acc.	0.849	0.827	0.840
	Cover.	**1.000**	**1.000**	**1.000**
W2V_GENERAL	Acc.	0.491	0.400	0.454
	Cover.	0.958	0.988	0.970
GloVe_DOMAIN	Acc.	0.866	0.802	0.840
	Cover.	**1.000**	**1.000**	**1.000**
GloVe_GENERAL	Acc.	0.754	0.588	0.686
	Cover.	0.958	0.988	0.970

Table 3: Resturants 200 adjs lexicon results

On the restaurant-adjectives-test-set and laptop-adjectives-test-set we measure the polarity accuracy (when a lexicon assigns the correct polarity) and the coverage (when a lexicon contains a polarity for the requested word).

Tables 3 and 4 show the results for restaurants and laptops respectively. In the tables the accuracy measures how many word polarities have been correctly tagged from the ones present in each lexicon (i.e. out-of-vocabulary words are not taken as errors). The coverage measures how many words were present in each lexicon regardless of the tagged polarity. The experiment shows that the static lexicons like

[5] http://www.yelp.com/dataset_challenge

[6] http://snap.stanford.edu/data/web-Amazon.html

[7] Compared to the billion words datasets employed in other works

[8] http://spark.apache.org/mllib/

[9] Please, refer to the Apache Spark Mllib Word2Vec documentation to see which the default parameters are

[10] With context-independent polarity we refer to those adjectives with unambiguous polarity not depending on the domain aspect they are modifying (e.g. *superb* is likely to be always positive, while *small* could be positive or negative depending on the context)

[11] Available at https://dl.dropboxusercontent.com/u/7852658/files/restaur_adjs_test.txt and https://dl.dropboxusercontent.com/u/7852658/files/laptops_adjs_test.txt respectively

[12] http://alt.qcri.org/semeval2015/task12/

LAPTOPS 200 ADJ GOLD LEXICON				
Name		Posit.	Neg.	Overall
General Inquirer	Acc.	0.965	0.971	0.967
	Cover.	0.677	0.479	0.605
BingLiu	Acc.	**0.971**	**0.984**	**0.976**
	Cover.	0.827	0.863	0.840
SentiWordNet	Acc.	0.795	0.833	0.809
	Cover.	0.921	0.904	0.915
QWN-PPV	Acc.	0.895	0.661	0.814
	Cover.	0.829	0.767	0.805
NRC_CANADA	Acc.	0.890	0.712	0.825
	Cover.	**1.000**	**1.000**	**1.000**
PMI_WINDOW_5	Acc.	0.850	0.395	0.720
	Cover.	0.843	0.589	0.750
W2V_DOMAIN	Acc.	0.874	0.740	0.825
	Cover.	**1.000**	**1.000**	**1.000**
W2V_GENERAL	Acc.	0.540	0.575	0.553
	Cover.	0.992	**1.000**	0.995
GloVe_DOMAIN	Acc.	0.890	0.740	0.835
	Cover.	**1.000**	**1.000**	**1.000**
GloVe_GENERAL	Acc.	0.849	0.589	0.754
	Cover.	0.992	**1.000**	0.995

Table 4: Laptops 200 adjs lexicon results

RESTAURANTS (SEMEVAL 2015 DATASET)					
Name		Prec.	Rec.	F1	Acc.
GI	posit.	0.783	0.937	0.853	0.760
	neg.	0.610	0.335	0.432	
BingLiu	posit.	0.810	**0.958**	**0.878**	**0.799**
	neg.	**0.731**	0.431	0.540	
SWN	posit.	0.790	0.896	0.840	0.745
	neg.	0.539	0.394	0.455	
QWN-PPV	posit.	0.751	0.954	0.841	0.733
	neg.	0.522	0.171	0.257	
NRC_CAN.	posit.	0.816	0.927	0.868	0.786
	neg.	0.648	0.471	0.546	
PMI_W_5	posit.	0.811	0.842	0.826	0.732
	neg.	0.493	0.503	0.498	
W2V_DOM	posit.	**0.848**	0.874	0.861	0.781
	neg.	0.582	**0.605**	**0.593**	
W2V_GEN	posit.	0.708	0.467	0.563	0.457
	neg.	0.228	0.488	0.311	
GloVe_DOM	posit.	0.792	0.940	0.860	0.770
	neg.	0.633	0.364	0.463	
GloVe_GEN	posit.	0.747	0.900	0.816	0.703
	neg.	0.404	0.210	0.277	

Table 5: Semeval 2015 restaurants results

GI and Liu's assign polarities with a very high precision, but they suffer from lower coverage. A similar behaviour can be observed for polarities based on WordNet. On the other hand the lexicons calculated directly on the domain datasets are less accurate, but they have much higher coverage. NRC_CANADA lexicon achieves a very good result, but it must be noted that it employs supervised information. The PMI based on windows achieve a quite good result despite of its simplicity, but it does not cover all the words (i.e. some words do not co-occur in the same context). The lexicons based on word embeddings calculated on the domain achieve a 100% coverage, because they are modelling the whole vocabulary, and offer a reasonable precision. Word embeddings (both Word2Vec and Glove) calculated on general domain corpus still cover a lot of the adjectives since they have been trained on a very large corpora, but they show a lower accuracy capturing the polarity of the words.

5.2. SemEval 2015 datasets based experiments

SemEval 2015 based datasets consists of quintuples of aspect-term, entity-attribute, polarity, and starting and ending position of the aspect-term. We are only interested in using the polarity slots, which refer to the polarity of a particular aspect of each sentence (not to the overall sentence polarity). We have applied the different lexicons to infer the polarity of each sentence, and then we have compared them to the gold annotations that come with the datasets.
The process of assigning a polarity to each sentence using the different polarity lexicons is the following:

- Only adjectives and verbs (e.g. *hate*, *recommend*) are taken into account to calculate polarity (common verbs like *be* and *have* are omitted)
- Negation words are taken into account to reverse the polarity of the subsequent word, in particular: *no, neither, nothing, not, n't, none, any, never, without*
- The number of positive and negative words according to each lexicon is counted. If the positives count is greater or equal to negatives count, the polarity of all polarity slots of the sentence is assigned as positive; and negative otherwise.

Notice that this is a very naive polarity annotation process. It is not intended to obtain good results but for comparing the lexicons against real sentences using the same setting. That is way in general the results are lower than in the experiment with the bare adjective lists. This naive polarity annotation process is repeated for every polarity lexicon so the different lexicons and methods can be compared under the same conditions in real reviews test sets.
Table 5 shows the results for restaurants dataset while table 6 shows the results for laptops dataset. These results have been calculated using the evaluation script provided by the SemEval 2015 organizers during the competition[13]. The results show that there is no a clear winner, and the best performing lexicon vary depending on the domain. Some

[13]Available at `http://alt.qcri.org/semeval2015/task12/index.php?id=data-and-tools`

LAPTOPS (SEMEVAL 2015 DATASET)					
Name		Prec.	Rec.	F1	Acc.
GI	posit.	0.631	0.939	0.755	0.651
	neg.	0.751	0.328	0.456	
BingLiu	posit.	0.64	0.96	0.768	0.669
	neg.	**0.821**	0.343	0.484	
SWN	posit.	0.63	0.903	0.742	0.638
	neg.	0.671	0.345	0.456	
QWN-PPV	posit.	0.605	0.9411	0.736	0.614
	neg.	0.675	0.228	0.341	
NRC_CAN.	posit.	0.653	0.922	0.764	0.673
	neg.	0.75	0.409	0.529	
PMI_W_5	posit.	0.622	0.841	0.715	0.611
	neg.	0.58	0.366	0.449	
W2V_DOM	posit.	**0.728**	0.825	**0.774**	**0.708**
	neg.	0.673	**0.636**	**0.654**	
W2V_GEN	posit.	0.533	0.443	0.484	0.441
	neg.	0.362	0.5	0.42	
GloVe_DOM	posit.	0.59	**0.971**	0.734	0.604
	neg.	0.762	0.159	0.263	
GloVe_GEN	posit.	0.571	0.932	0.708	0.567
	neg.	0.528	0.120	0.196	

Table 6: Semeval 2015 laptops results

lexicon seem to be more accurate capturing positive words and others seem to have a better recall. It must be noted that in this case what is being annotated are whole sentences of actual reviews, so there are a lot of facts involved apart from the mere polarity of single words. Also in this case the domain-based word embeddings work better capturing the polarity than their general-domain counterparts.

6. Conclusions

In this work we have compared different existing lexicons and methods to obtain a polarity value for words in a particular domain. We have shown a simple yet functional way to quickly get a polarity value only with unlabelled texts using continuous word representations. It is similar in essence to other exiting methods that require co-occurrence computations among words, but the semantic properties of the continuous word embeddings does not require words to co-occur and is easier to compute. In addition we have shown that the similarity of sentiment bearing words (mainly adjectives) is better modelled using a smaller in-domain dataset rather than a bigger general dataset. We have observed a similar behaviour in preliminary experiments for other languages such us Spanish, French or Italian. An obvious advantage is that provided enough unlabelled domain data, the word embeddings and polarity scores can be easily obtained for any language. As a further work, we would like to experiment with these in-domain calculated word embeddings (and other variants) within more complex sentiment analysis systems to see if they improve the performance. Also, many machine learning based sentiment analysis approaches in the literature already employ word embeddings as input features, usually computed against very big general corpora. It would be interesting to see how general domain word embeddings, which provide general language knowledge, and in-domain calculated word embeddings, which provide domain-aware information, can be combined to improve the results of such systems. Also we would like to explore if approaches with a more weakly supervised nature, like topic modelling and Latent Dirichlet Allocation based systems, that try to jointly model the polarity and other facets of documents could benefit from the information coming from in-domain word embeddings.

7. Acknowledgements

This work has been supported by Vicomtech-IK4 and partially funded by TUNER project (TIN2015-65308-C5-1-R).

8. Bibliographical References

Agerri, R. and Garcia, A. (2009). Q-WordNet : Extracting Polarity from WordNet Senses. *Seventh Conference on International Language Resources and Evaluation Malta Retrieved May*, 25:2010.

Baccianella, S., Esuli, A., and Sebastiani, F. (2010). SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining. *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, 0:2200–2204.

Bengio, Y., Ducharme, R., Vincent, P., and Janvin, C. (2003). A Neural Probabilistic Language Model. *The Journal of Machine Learning Research*, 3:1137–1155.

Brody, S. and Elhadad, N. (2010). An unsupervised aspect-sentiment model for online reviews. *The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, (June):804–812.

Dumais, S., Furnas, G., Landauer, T., Deerwester, S., Deerwester, S., et al. (1995). Latent semantic indexing. In *Proceedings of the Text Retrieval Conference*.

Dumais, S. T. (2004). Latent semantic analysis. *Annual review of information science and technology*, 38(1):188–230.

Esuli, A. and Sebastiani, F. (2006). SentiWordNet : A Publicly Available Lexical Resource for Opinion Mining. *Proceedings of LREC 2006*, pages 417–422.

Fellbaum, C. (1998). *WordNet*. Wiley Online Library.

García-Pablos, A., Cuadros, M., and Rigau, G. (2015). Unsupervised word polarity tagging by exploiting continuous word representations. *Procesamiento del Lenguaje Natural*, 55:127–134.

Guerini, M., Gatti, L., and Turchi, M. (2013). Sentiment Analysis : How to Derive Prior Polarities from SentiWordNet. *Emnlp*, pages 1259–1269.

Hatzivassiloglou, V. and McKeown, K. R. (1997). Predicting the semantic orientation of adjectives. *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the*

European Chapter of the Association for Computational Linguistics, pages:181.

Hill, F., Cho, K., Jean, S., Devin, C., and Bengio, Y. (2014). Embedding Word Similarity with Neural Machine Translation. pages 1–12.

Hu, M. and Liu, B. (2004). Mining opinion features in customer reviews. *AAAI*.

Huang, E. H., Socher, R., Manning, C. D., and Ng, A. (2012). Improving word representations via global context and multiple word prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 873–882.

Iacobacci, I., Pilehvar, M. T., and Navigli, R. (2015). SensEmbed: Learning Sense Embeddings for Word and Relational Similarity. *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, (1):95–105.

Jaggi, M., Zurich, E. T. H., and Cieliebak, M. (2014). Swiss-Chocolate : Sentiment Detection using Sparse SVMs and Part-Of-Speech n -Grams. (SemEval):601–604.

Ji, S., Yun, H., Yanardag, P., Matsushima, S., and Vishwanathan, S. V. N. (2015). WordRank: Learning Word Embeddings via Robust Ranking. pages 1–12.

Kim, J.-k. (2013). Deriving adjectival scales from continuous space word representations. *Emnlp*, (October):1625–1630.

Kiritchenko, S., Zhu, X., Cherry, C., Mohammad, S. M., and Mohammad, S. (2014). NRC-Canada-2014 : Detecting Aspects and Sentiment in Customer Reviews. (SemEval):437–442.

Le, Q. and Mikolov, T. (2014). Distributed Representations of Sentences and Documents. *International Conference on Machine Learning - ICML 2014*, 32:1188–1196.

Li, J. and Jurafsky, D. (2015). Do multi-sense embeddings improve natural language understanding? *arXiv preprint arXiv:1506.01070*.

Lin, C., Road, N. P., and Ex, E. (2009). Joint Sentiment / Topic Model for Sentiment Analysis. *Cikm*, pages 375–384.

Maks, I., Izquierdo, R., Frontini, F., Agerri, R., Azpeitia, A., and Vossen, P. (2014). Generating Polarity Lexicons with WordNet propagation in five languages. pages 1155–1161.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient Estimation of Word Representations in Vector Space. *arXiv preprint arXiv:1301.3781*, pages 1–12, January.

Mikolov, T., Le, Q. V., and Sutskever, I. (2013b). Exploiting Similarities among Languages for Machine Translation. In *arXiv preprint arXiv:1309.4168v1*, pages 1–10.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J. (2013c). Distributed Representations of Words and Phrases and their Compositionality. *arXiv preprint arXiv: ...*, pages 1–9, October.

Mikolov, T., Yih, W.-t., and Zweig, G. (2013d). Linguistic regularities in continuous space word representations. *Proceedings of NAACL-HLT*, pages 746–751.

Pang, B. and Lee, L. (2008). Opinion mining and sentiment analysis. *Foundations and trends in information retrieval*, 2(1-2):1–135.

Pavlopoulos, J. and Androutsopoulos, I. (2014). Aspect term extraction for sentiment analysis: New datasets, new evaluation measures and an improved unsupervised method. *Proceedings of LASMEACL*, pages 44–52.

Pennington, J. and Manning, C.). Glove: Global vectors for word representation. *Emnlp2014.Org*.

Popescu, A. and Etzioni, O. (2005). Extracting product features and opinions from reviews. *Natural language processing and text mining*, (October):339–346.

Qiu, G., Liu, B., Bu, J., and Chen, C. (2011). Opinion word expansion and target extraction through double propagation. *Computational linguistics*, (July 2010).

Ramos, C. and Marques, N. C. (2005). Determining the Polarity of Words through a Common Online Dictionary.

Rothe, S., Ebert, S., and Schütze, H. (2016). Ultradense word embeddings by orthogonal transformation. *arXiv preprint arXiv:1602.07572*.

Schwartz, R., Reichart, R., and Rappoport, A. (2014). Symmetric Pattern Based Word Embeddings for Improved Word Similarity Prediction. 353.

Socher, R., Perelygin, A., Wu, J., and Chuang, J. (2013). Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. *newdesign.aclweb.org*.

Stone, P. J., Dunphy, D. C., and Smith, M. S. (1966). The general inquirer: A computer approach to content analysis.

Taboada, M., Brooke, J., Tofiloski, M., Voll, K., and Stede, M. (2011). Lexicon-Based Methods for Sentiment Analysis. *Computational Linguistics*, 37(September 2010):267–307.

Tang, D., Wei, F., Yang, N., Zhou, M., Liu, T., and Qin, B. (2014a). Learning Sentiment-Specific Word Embedding. *Acl*, pages 1555–1565.

Tang, D., Wei, F., Yang, N., Zhou, M., Liu, T., and Qin, B. (2014b). Learning sentiment-specific word embedding for twitter sentiment classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 1555–1565.

Turian, J., Ratinov, L., and Bengio, Y. (2010). Word Representations: A Simple and General Method for Semi-supervised Learning. *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 384–394.

Turney, P. D. (2002). Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews. *Computational Linguistics*, (July):8.

Vicente, S., Agerri, R., and Rigau, G. (2014). Simple , Robust and (almost) Unsupervised Generation of Polarity Lexicons for Multiple Languages. *Eacl2014*.

Zhang, L. and Liu, B. (2014). Aspect and entity extraction for opinion mining. In *Data mining and knowledge discovery for big data*, pages 1–40. Springer.