Procesamiento del Lenguaje Natural, Revista nº 55, septiembre de 2015, pp 127-134 recibido 27-03-2015 revisado 26-04-2015 aceptado 15-05-2015

Unsupervised Word Polarity Tagging by Exploiting Continuous Word Representations

Etiquetado no supervisado de la polaridad de las palabras utilizando representaciones continuas de palabras

Aitor García-Pablos, Montse Cuadros
Vicomtech-IK4 research centre
Mikeletegi 57, San Sebastian, Spain
{agarciap,mcuadros}@vicomtech.org

German Rigau
IXA Group
Euskal Herriko Unibertsitatea
San Sebastian, Spain
german.rigau@ehu.es

Resumen: El análisis de sentimiento es un campo del procesamiento del lenguaje natural que se encarga de determinar la polaridad (positiva, negativa, neutral) en los textos en los que se vierten opiniones. Un recurso habitual en los sistemas de análisis de sentimiento son los lexicones de polaridad. Un lexicón de polaridad es un diccionario que asigna un valor predeteminado de polaridad a una palabra. En este trabajo exploramos la posibilidad de generar de manera automática lexicones de polaridad adaptados a un dominio usando representaciones continuas de palabras, en concreto la popular herramienta Word2Vec. Primero mostramos una evaluación cualitativa de la polaridad sobre un pequeño conjunto de palabras, y después mostramos los resultados de nuestra competición en la tarea 12 del SemEval-2015 usando este método.

Palabras clave: word embeddings, polaridad de palabras, análisis de sentimiento

Abstract: Sentiment analysis is the area of Natural Language Processing that aims to determine the polarity (positive, negative, neutral) contained in an opinionated text. A usual resource employed in many of these approaches are the so-called polarity lexicons. A polarity lexicon acts as a dictionary that assigns a sentiment polarity value to words. In this work we explore the possibility of automatically generating domain adapted polarity lexicons employing continuous word representations, in particular the popular tool Word2Vec. First we show a qualitative evaluation of a small set of words, and then we show our results in the SemEval-2015 task 12 using the presented method.

Keywords: word embeddings, word polarity, sentiment analysis

1 *Introduction*

During the last decade the online consumer opinions have become a very valuable resource of information for companies. The huge amount of user generated content containing opinions about products, services and virtually about everything, requires automatic processing tools to handle all this data. Sentiment Analysis is the field of Natural Language Processing (NLP) that focus on determining the sentiment contained in opinion texts(Liu, 2012).

The determination of the sentiment in a text usually consists of finding subjective sentences or expressions and classifying them inside one of the possible sentiment values. Regardless if the sentiment is a continuous value or a categorical label (e.g. positive, very positive, negative, neutral, etc.), one of the key challenges in Sentiment Analysis is how to determine it for the words observed in the text under analysis.

There are many different approaches in the literature: some of them employ supervised machine learning methods to train a model that learns which words/expressions/sentences are positives and which are negatives. Other methods rely on sentiment lexicons, which are dictionaries manually developed or bootstrapped from texts using different techniques. A sentiment lexicon is an important tool for many Sentiment Analysis techniques. They can be used standalone as the only indicator for the polarity of a word, or as an additional feature for more sophisticated methods. Sentiment lexicons are domain dependent, meaning that some words or expressions may vary their po-

ISSN 1135-5948

larity from one domain to another(Choi and Cardie, 2009). If a process to create a Sentiment Lexicon is too complex or too time consuming it would be difficult to port to new domains and languages. In this paper we propose a simple yet promising approach employing continuous word representations to obtain domain-aware polarity for words.

The rest of the paper is structured as follows. Section 2 introduces previous work on deriving word polarities for Sentiment Analysis. Section 3 describes our proposed approach to obtain a polarity value for words using a continuous word embedding model, in particular by exploiting Word2Vec. Section 4 shows the results of our first experiments, in particular some qualitative analysis of adjectives in three different domains, and the results of our participation in the SemEval-2015 task 12 competition using the method exposed in this paper to calculate words polarity. Finally, section 5 contains our conclusions and future work.

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).

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, Se-bastiani and Moruzzi (2006) the authors em-ploy the glosses that accompany each Word-Net synset[1] to perform a semi-supervised synset classification. The result consists of three scores per synset: positivity, negativ-ity and objectivity. In Baccianella, Esuli and Sebastiani (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, Gatt, and Turchi (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, Agerri, and Rigau, 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 et al., (1997) the authors model the corpus as a graph of adjectives joined by con-junctions. 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 adjec-tives joined by ”but” will have opposite ori-entations.

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

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

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, Ratinov, and Bengio, 2010; Huang et al., 2012; Mikolov et al., 2013b). 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., 2014; Pavlopoulos and Androutsopoulos, 2014). In Kim (2013) the word embeddings are explored to derive adjectival scales.

In this work we employ word embeddings, in particular the popular Word2Vec tool (Mikolov et al., 2013a; Mikolov, Yih, and Zweig, 2013) to obtain a polarity value for each word. As the Word2Vec model is trained on a corpus of the target domain it should be able to capture domain specific semantics, and in this case, domain specific polarities. The method is rather simple and its unsupervised nature makes it easy to apply to new languages or domains given a big enough text corpus.

3 Our approach

Our aim is to assess whether continuous word representations can be leveraged to automatically infer the polarity of the words for a given domain, just employing unlabeled text from the domain (for example, customer reviews) and a minimal set of seed words. The intuition behind this idea is based on the following assumptions:

- Continuous word representations, also called word embedding, can capture the semantic similarity between words.
- The polarity of a new given word with unknown polarity can be established by simply measuring its relative similarity with respect to two small seed sets of known positive and negative words from the domain (and its associated word embeddings).
- This fact can be exploited to arrange all the words in the vocabulary into a positive-negative axis.

Continuous word representations are mappings between entries in a vocabulary (i.e. the words) and numeric vectors of a certain size that represent the words. Depending on how these vectors are computed different linguistic or semantic facets would be captured. Albeit there are many different ways in the literature to obtain such vector representations, our experiments are based on Word2Vec[2]. Word2Vec is known to capture certain semantic patterns quite effectively, from semantically related words (e.g. obtaining "France", "Italy" and "Portugal" as similar words to the word "Spain") to more complex analogy patterns (e.g. "king" is to "man" which "queen" is to "woman").

Of course the performance and the kind of results that can be expected largely depend on the corpus used for training. We employ a domain specific dataset to obtain polarity values for a specific domain.

3.1 Datasets

To generate the Word2Vec word embeddings we have used datasets from different domains. The first dataset consists of customer reviews about restaurants. It is a 100k review subset obtained from the Yelp dataset[3]. From now on we will refer to it as Yelp-restaurants.

We also have used a second dataset of customer reviews about laptops. This dataset contains a subset of about 100k reviews from the Amazon electronic device review dataset from the Stanford Network Analysis Project

[2] We use the Word2Vec implementation contained in the Apache Spark Mllib library with its default parameters: vector size 100, skip-grams with context window 5, learning rate 0.025. `https://spark.apache.org/mllib/`

[3] `http://www.yelp.com/dataset_challenge`

(SNAP)[4], selecting reviews that contain the word "laptop".

In addition to these English datasets, we have also used the Spanish film reviews dataset from MuchoCine (Cruz et al., 2008) which contains about 4k reviews written in Spanish.

3.2 Generating the model

Word2Vec works processing plain text, taking every white-space separated token as a word. It builds a vocabulary with all the different word forms found in the training corpus. It is usual to set a minimum frequency threshold to discard those words that appear less than a certain number of times.

Before starting the process we perform a pre-processing step consisting of tokenizing, Part-of-Speech tagging and lemmatizing the datasets. For this pre-processing we use the IXA-pipes toolkit for both English and Spanish reviews [5].

Lemmatization of terms helps reducing word sparsity, because our datasets are not as big as the ones used in the literature. Part-of-Speech tagging serves to filter out non-content words (e.g. all determiners, pronouns, etc.). For other semantic tasks keeping every word might be necessary, but for our polarity-calculation task we only need content-words (nouns, verbs, adjectives and adverbs).

Once we have the dataset pre-processed, it takes only a few minutes on a commodity desktop computer to obtain the semantic word vectors using Word2Vec.

Using the Word2Vec vectors we can assign a polarity value to each word from the domain using the following simple equation:

$$\text{polarity}(w) = \text{sim}(w, POS) - \text{sim}(w, NEG) \quad (1)$$

Where POS is a set of known positive words for the domain of interest, and analogously NEG is a set of known negative words. In our experiments we have used domain independent words, like *excellent* and *horrible* respectively, or their equivalents in other languages (e.g. *excelente* and *horrible* in Spanish). We have used the cosine distance as a similarity function *sim* between the computed semantic vectors.

In that way we can obtain a continuous polarity value for every word in the domain. This word would be positive if the similarity between the target word and POS is greater than the similarity between that same word and NEG, and vice versa. 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.

3.3 Dealing with multiword terms

Handling multiword terms is important in Sentiment Analysis systems (e.g. it is not the same to detect just "memory" than "flash memory" and/or "RAM memory", etc.). It is also important for tasks like opinion target detection or in this case, to better detect the sentiment bearing words. For example, in Spanish the word "pena" (sadness) would probably be taken as a negative word, but the expression "merecer la pena" (be worth) has the opposite polarity. Multiword terms can be also found as opinion expressions like "top notch" or "blazing fast". Finally, multiword terms arise from usual collocations of single terms, so they vary between domains.

Multiword terms are expressions that are formed by more than a single word, like idioms, typical expressions or usual word collocations. Multiword terms depend on the language and also on the topic or domain of the analysed texts. For example, in restaurant domain it is very common to find multiword terms related to recipes or names of dishes and ingredients (e.g. "black cod", "spring roll", "orange juice"). In the computer domain we have multiword terms for components like "RAM memory", "hard disk", "touch pad", "graphics card", "battery life", etc.

Handling multiword terms in advance is the only way to let Word2Vec indexing them as a vocabulary entry. Without multiword terms pre-processing it would not be possible to query the model for the polarity of expressions like "top notch", "high quality" or "high resolution", because their composing words would have been treated individually.

In order to bootstrap a list of candidate domain related multiword terms we have computed the Log-Likelihood Ratio (LLR)

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

[5] http://ixa2.si.ehu.es/ixa-pipes/

Restaurants	Laptops
happy hour	tech support
onion ring	power supply
ice cream	customer service
spring roll	operating system
live music	battery life
wine list	signal strength
filet mignon	sound quality
goat cheese	plug and play
bread sticks	numeric keypad

Table 1: Examples of multiword terms obtained for restaurants and laptop domains.

of word n-grams (with n<=3) to detect the more salient word collocations. Then we keep the top K candidates from the list ranked by the LLR measure. LLR is a common measure in the literature to estimate if two events (two words in this case) co-occur by chance or if they are truly correlated. In the case of word n-grams with n>2, the LLR is calculated taking the first word of the n-gram as the first event, and the rest of the n-1 words atomically as the second event.

With no other processing this leads to a very noisy list, in which many candidate collocations are formed by stopwords (determiners, pronouns, and other undesired words). To prevent this we first analyze the corpus to obtain the Part-of-Speech tags of the words. Then we run the calculation of the LLR for all the word n-grams in the text as before, but we keep the Part-of-Speech information of every word that composes a candidate multiword. Using the Part-of-Speech information of individual words that compose the candidate multiwords we filter out the ones that do not follow certain desired patterns (e.g. noun+noun, adj+noun, noun+prep+noun, etc.).

Table 1 shows some examples of the obtained multiword terms for the restaurant and laptop domains.

4 Experiments and results

In this section we present some preliminary results and propose ideas for further experimentation.

4.1 Qualitative tests

We have trained three Word2Vec models and we have generated some lexicons for a small set of highly frequent opinion words and expressions from each domain and corpus. More precisely we have manually annotated 200 adjectives taken from the dataset of each domain.

Word	Pol. Score	Pol. label
delicious	0.4249	positive
tasty	0.4393	positive
inexpensive	0.3416	positive
top notch	0.2850	positive
lot of money	-0.3510	negative
slow	-0.1825	negative
arrogant	-0.2544	negative
mediocre	-0.0517	negative
fantastic	0.2408	positive
prompt	0.0766	positive
amazing	0.2659	positive
outstanding	0.1748	positive
fresh	0.3178	positive
terrible	-0.3065	negative
lousy	-0.0517	negative
poor	-0.2427	negative
yummy	0.2940	positive
pleasant	0.0112	positive
disappointing	-0.0591	negative
terrific	0.2641	positive
boring	-0.0438	negative
pathetic	0.1322	**positive***
nasty	-0.1636	negative

Table 2: Examples of polarity values obtained from the restaurants polarity lexicon.

The first two models correspond to English restaurant reviews and laptop reviews. To train the Word2Vec models we have used Yelp-restaurants and Amazon-laptops datasets described in section 3.1 respectively.

Table 2 shows some results for adjectives in the restaurant domain. Table 3 shows some results for adjectives in the laptop domain. About 80% of the 200 manually annotated adjectives for restaurant domain are correctly annotated. For laptop domain the 70% of the 200 manually annotated adjectives are correctly annotated. Some examples of polarity values that seem incorrect or counter-intuitive are marked in bold with an asterisk.

In addition we have trained another model using Spanish movie reviews from MuchoCine corpus. The process to generate the Word2Vec model is the same and the only thing that must be adapted to calculate the polarity are the POS and NEG words. For example, instead of *excellent* and *horrible* we have translated them to their equivalents in Spanish, *excelente* and *horrible*.

Table 4 show some results for Spanish adjectives on the films domain. It seems that

Word	Pol. Score	Pol. label
slow	-0.0790	negative
fast	0.2007	positive
quick	0.1605	positive
crappy	-0.1753	negative
great	0.4287	positive
nice	0.2884	positive
old	-0.0387	negative
modern	0.0590	positive
glossy	0.0786	positive
top notch	0.3245	positive
incredible	0.2852	positive
funny	-0.3196	**negative***
pricey	0.0415	**positive***
bug	-0.2780	negative
break	-0.3196	negative
futuristic	0.0027	positive
trendy	0.1818	positive
high resolution	0.2122	positive
high quality	0.2069	positive
nothing but praise	-0.1318	negative
lot of problem	-0.3865	negative

Table 3: Examples of polarity values obtained from the laptops polarity lexicon.

Word	Pol. Score	Pol. label
bonito	0.1008	positive
bueno	0.6570	positive
fabuloso	0.4191	positive
increíble	0.3452	positive
inolvidable	0.3368	positive
fantástico	0.3929	positive
divertir	0.4111	positive
alucinante	-0.1228	**negative***
aburrir	-0.0501	negative
repetitivo	-0.1220	negative
absurdo	-0.0509	negative
estúpido	-0.0308	negative
brillante	0.7182	positive
genial	0.6745	positive
asombroso	-0.0625	**negative***
atractivo	0.3001	positive
enfermizo	-0.0708	negative
simple	-0.0708	negative
carismático	0.3336	positive
profundo	-0.3846	**negative***
deleznable	0.0712	**positive***

Table 4: Examples of polarity values obtained from the movies polarity lexicon (for Spanish).

the polarities are less accurate than in the English tests. Obviously, one possible reason could be the use of a much smaller corpora (4k film reviews vs. 100k for restaurants and laptops) which could be simply too small to obtain an accurate model. We leave this for future experiments.

4.2 Experiments at SemEval-2015

Finally, in order to perform a more systematic experiment to assess the validity of the polarities, we participated in the SemEval-2015 task 12 about Aspect Based Sentiment Analysis (ABSA) using this approach to calculate the polarity of words. Two training datasets were provided. 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). The annotation consists of quintuples of aspect-term, entity-attribute, polarity, and starting and ending position of the aspect-term. Since our method is unsupervised, we did not use the training datasets.

For the competition in the polarity annotation subtask similar datasets were provided, with the polarity slot empty. The subtask was about filling the polarity slot of each quintuple. For every sentence we performed the polarity annotation just counting positive and negative words (according to the Word2Vec polarity calculation) and assigning the most frequent polarity to the polarity slot of each quintuple. We took into account the negation words present in the sentence in order to reverse the polarity of the words within a certain window (one token before and two tokens after the current word). In particular, the negation words employed were: *not, neither, nothing, no, none, any, never, without, cannot.*

Table 5 shows the accuracy results for restaurant and laptop domain as they were reported in the competition. The table also shows the result of the best performing system in the competition for that subtask, the average score of all 14 participant systems and the baselines provided by the SemEval organizers. The SVM+BOW baseline is a supervised baseline that employs a Support Vector Machine based training and classification using a Bag-of-Words approach as features. The Majority baseline assigns the most frequent polarity in the training dataset. To our knowledge, best performing systems from other participants were supervised approaches trained on the provided training datasets.

Our system was performing a very basic

Polarity	Restaurants acc.	Laptops acc.
SVM+BOW	0.635	0.699
Majority	0.537	0.570
Our system	0.694	0.683
Best system	0.786	0.793
Average	0.713	0.713

Table 5: Polarity annotation accuracy results on the restaurant, laptops for slot 3.

and naive polarity annotation, relying only in the polarity values given by our trained Word2Vec model, but in our opinion the results are quite promising.

5 Conclusions and future work

In this paper we have described our approach with continuous word representations to calculate a polarity value for words for any domain and language. We use the popular Word2Vec tool to compute a vector model for the words coming from datasets of different domains. Having the appropriate corpora and seed words, this approach could provide a domain-specific lexicon with polarities for any language. As a first test, we perform a qualitative observation of the polarity values for three different domains and in two different languages. Then, we presented the results obtained in the SemEval-2015 task 12 competition, in the polarity annotation subtask, achieving quite good results despite the simple and unsupervised nature of the approach. The idea introduced in this work requires further research to assess if the method works for other domains and languages. Additional investigation is required on the effect of different parameters (dimensionality of Word2Vec vectors, number of training iterations, size of context window, size of corpora, etc.).

Acknowledgements

This work has been supported by Vicomtech-IK4.

References

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

Baccianella, S., A. Esuli, and F. Sebastiani. 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., R. Ducharme, P. Vincent, and C. Janvin. 2003. A Neural Probabilistic Language Model. *The Journal of Machine Learning Research*, 3:1137–1155.

Brody, S. and N Elhadad. 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.

Choi, Y. and C. Cardie. 2009. Adapting a polarity lexicon using integer linear programming for domain-specific sentiment classification. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2-Volume 2*, pages 590–598. Association for Computational Linguistics.

Cruz, F. L, J. A. Troyano, F. Enriquez, and J. Ortega. 2008. Clasificación de documentos basada en la opinión: experimentos con un corpus de crıticas de cine en espanol. *Procesamiento de Lenguaje Natural*, 41.

Dumais, S, G Furnas, T Landauer, S Deerwester, S Deerwester, 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., F. Sebastiani, and V. G. Moruzzi. 2006. SentiWord-Net : A Publicly Available Lexical Resource for Opinion Mining. *Proceedings of LREC 2006*, pages 417–422.

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

Guerini, M., L. Gatti, and M. Turchi. 2013.

Sentiment Analysis : How to Derive Prior Polarities from SentiWordNet. *Emnlp*, pages 1259–1269.

Hatzivassiloglou, V., V. Hatzivassiloglou, K.R. McKeown, and K.R. McKeown. 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.

Huang, E. H, R. Socher, C. D. Manning, and A. Ng. 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.

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

Liu, B. 2012. Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies*, 5(1):1–167.

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

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

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

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

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

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

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

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

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

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

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

Tang, D., F. Wei, N. Yang, M. Zhou, T. Liu, and B. Qin. 2014. 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., L. Ratinov, and Y. Bengio. 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 Orienta-tion Applied to Unsupervised Classifica-tion of Reviews. *Computational Linguis-tics*, (July):8.

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

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