Sentiment-specific Word Embeddings with Application for Sentiment Classification in Short Texts

Matija Piškorec
Division of Electronics
Ruđer Bošković Institute
Zagreb, Croatia
matija.piskorec@irb.hr

***Abstract*—In this seminar I will give an overview of several word embeddings methods used in natural language processing, with an emphasis on sentiment analysis of short texts. Finding an appropriate word embeddings automatically avoids task-specific engineering of textual features and usually results in methods that are more versatile and that can perform in wide array of language processing tasks. Many of these methods are based on neural network language models, especially convolutional neural network architectures. In this seminar I give an overview of two such methods - Collobert and Westons (C&W) model that leverages syntactic context of words, and sentiment specific-word embeddings (SSWE) model that leverages sentiment polarity of the text. Along with these, I also give a short overview of other related distributed representation methods, with special emphasis on those that use convolutional neural network architecture for sentiment analysis. In the end, I describe Several recent sentiment analysis tasks of SemEval challenges which served as an exemplary benchmark for identifying state-of-the-art methods in the field of sentiment analysis. These challenges demonstrated that methods which use word embeddings are competitive with the methods which use manually engineered features.**

I. Introduction

Standard approach for sentiment analysis is to use one of the supervised [1] or unsupervised [2] machine learning techniques on annotated text corpus. First datasets were user-generated reviews collected from various websites, for example Epinions [2] or Internet movie database (IMDb) [1]. In these cases a reasonable assumptions was that the sentiment of a review is quantified with the final score assigned to it. In recent years the emphasis shifted to texts where these kinds of annotations are not available. The most prominent example is Twitter - a free service where users can share short (limited to 160 characters) messages, called *Tweets*, with their *followers*. The shortness of Tweets makes it difficult even to express, let alone infer, their sentiment. Nevertheless, due to Twitter's popularity it generated a lot of research interest [3], [4], including a very popular machine learning challenge [5] (which I describe in more detail in section III-B).

One notable difference between sentiment analysis of reviews and sentiment analysis of Tweets is that reviews typically have a specific target toward which the sentiment is expressed - a movie being reviewed. In comparison, Tweets are general texts which can contain sentiments towards multiple targets. One approach is to identify all potential targets in a tweet and to use target-dependent features to identify sentiment toward each of them [3].

II. Word vector representation for semantic analysis

Idea to use vector word representations [6] to elicit semantics from text is not new, one of the earliest approaches is Latent Semantic Analysis (LSA) which learns semantic word vectors by singular value decomposition (SVD) of a term-document co-occurrence matrix [7]. One of the word vector representations specifically developed for sentiment analysis is word vectors for sentiment analysis (WVSA) method [8], which allows learning word vectors using an unsupervised probabilistic model of documents. This approach is similar to Latent Dirichlet Allocation (LDA) [9], except that it aims to model word representations instead of latent topics.

A powerful approach to learning word representations [1] is with neural networks [10]. However, first neural language models were very inefficient due to the large number of parameters which needed to be optimized. However, this changed in recent years due to the development of efficient optimization methods which allowed training of neural networks with many hidden layers, commonly referred to as *deep learning* [11]. One of currently popular approaches is to use skip-gram model [2] instead of hidden nonlinear layer, which greatly increases efficiency of learning [12]. This approach generates vector representations on which it is possible to perform simple semantic algebraic operations. For example, if we subtract representation of "Spain" from representation for "Madrid", and add representation for "France", we will obtain representation which is very close in vector space to "Paris", which means that these representations are able to capture semantic relationship between these words [13].

[1] Another term often used in literature, along with word vector representation, is *word embeddings*. In neural network literature it is common to use *distributed representations*.

[2] Skip-gram model takes one specific word and then tries to predict the surrounding words.

However, the most ambitious task is to use neural networks as an architecture to solve all common natural language processing tasks such as parts-of-speech tagging, named entity recognition and semantic role labeling. For a good overview see [14]. One of the approaches is Collobert and Weston (C&W) model [15], [16] which leverages syntactic context of words, and which I briefly describe in section II-A. C&W model is used as a starting point for the sentiment specific-word embeddings (SSWE) model [17], [18] which uses similar approach to leverage sentiment polarity of text. I describe it in section II-B. I also give an overview of other state-of-the-art methods that use convolutional neural network architecture for sentiment analysis in section II-C.

A. Collobert and Weston (C&W) model

C&W model [15], [16] was developed as a common approach to solving all common natural language processing tasks using only raw texts as an input, with minimal preprocessing (just lowercasing and encoding capitalization as an additional feature). They choose following four tasks as benchmarks: parts-of-speech tagging (POS), chunking (CHUNK), named entity recognition (NER) and semantic role labeling (SLR), and were able to achieve performance close to the state-of-the-art methods [16].

C&W model consists of a neural network architecture with a *lookup table*, two linear and a HardTanh nonlinear layer [15], [16]. The architecture of the network is designed to extract very generic features present in the text. Inputs are sequences of words, rather than individual words, which enforces generation of features that that capture local information - the context. Similar approach is used in learning image representations, where convolution layers are used to generate features capturing local information.

Language model score $f^{cw}(t)$ for an input ngram t is calculated in the following way. First, lookup table entry L_t for the ngram t is passed through the first linear layer with the weights w_1 (including a bias weight b_1) and an HardTanh activation function:

$$a = HardTanh(w_1 L_t + b_1) \tag{1}$$

With HardTanh defined as:

$$HardTanh(x) = \begin{cases} -1 & \text{if } x < -1 \\ x & \text{if } -1 \leq x \leq 1 \\ 1 & \text{if } x > 1 \end{cases} \tag{2}$$

Second, output is passed through the second linear layer, producing language model score $f^{cw}(t)$:

$$f^{cw}(t) = w_2(a) + b_2 \tag{3}$$

Model is trained in unsupervised way. The training objective is to optimize a hinge loss between an original ngram t and a corrupted ngram t^r where a middle word is replaced by a random word [15]:

$$loss_{cw}(t, t^r) = max(0, 1 - f^{cw}(t) + f^{cw}(t^r)) \tag{4}$$

B. Sentiment specific-word embeddings (SSWE) model

The original SSWE paper [17] describes three SSWE models: $SSWE_h$ and $SSWE_r$ that learn sentiment-specific word embeddings, and an unified model $SSWE_u$ that learns embeddings based on both sentiment and syntactic context of words. Here I explain just the unified model $SSWE_u$, as it is the one on which the Coooolll system is based [18]. In addition to the syntactic part of the loss function $loss_{cw}$, $SSWE_u$ defines an additional sentiment hinge loss $loss_{us}$ between an original ngram t and a corrupted ngram t^r:

$$loss_{us}(t, t^r) = max(0, 1 - \delta_s f^u(t) + \delta_s f^u(t^r)) \tag{5}$$

Where $\delta_s(t)$ is an indicator function which codes the sentiment polarity of a sentence (1 for a positive sentiment and -1 for a negative sentiment). The final loss function of $SSWE_u$ is a linear combination between the two hinge losses, with parameter α weighting their influence:

$$loss_u(t, t^r) = \alpha \cdot loss_{cw}(t, t^r) + (1 - \alpha) \cdot loss_{us}(t, t^r) \tag{6}$$

Note that the syntactic loss function $loss_{cw}$ in equation 4 enforces that the true ngram t has a higher language model score f^w that the corrupted ngram t^r, while sentiment loss function $loss_{us}$ in equation 4 enforces that the true ngram t is more consistent with the true sentiment annotation than the corrupted ngram t^r.

The training of $SSWE_u$ model is performed through backpropagation using adaptive subgradient method (AdaGrad) [19]. Embeddings for unigrams, bigrams and trigrams were learned separately. They dataset on which they train consists of 10 million Tweets, 5 million with positive emoticons and 5 million with negative emoticons[3]. Using emoticons as a substitute for real sentiment annotations is called *distant supervised learning* [20]. I expect that better results could be achieved by more carefully labeling the sentiment of Tweets, maybe with popular crowdsourcing services like Amazon Turk (https://www.mturk.com/mturk/welcome) or by considering emojis which allow much richer expression of sentiment than emoticons [21].

The choice of hyperparameters is performed through experimentation - for example, a range of $[0.5, 0.6]$ for the parameter α. Too low or too large values of α give worst performance, highlighting the importance of both syntactic and sentiment context for sentiment classification.

After training of individual unigram, bigram and trigram embeddings it is necessary to combine them to obtain final Tweet embeddings. For this they apply *min*, *max* and *average* convolutional operators on the ngram embeddings and concatenate the results.

The embeddings produced by $SSWE_u$ model achieve better performance on sentiment classification of Tweets than pure C&W model, which is understandable considering that C&W

[3]Positive emoticons are :) :) :-) :D =), while negative emoticons are :(: (:-([4].

model is poor at discriminating between syntactically very similar words like "good" and "bad" which have diametrically opposing sentiment associations. This is true for most other word embedding representations which do not use sentiment information directly, for example skip-gram model [12], [12] which I mentioned in section II.

C. Other deep convolutional neural networks models for sentiment analysis

Over the past few years there was a lot of research on neural network architectures and their application in sentiment analysis. Majority of these are based on deep convolutional neural network architecture and share much similarity with C&W model, SSWE model being just one of them. In this section I give an overview of the related approaches.

In [22] authors describe CharSCNN architecture - a deep convolutional neural network for sentiment analysis of short texts. Their model is conceptually similar to the C&W model, with an addition of one convolutional layer which extracts character features. For evaluation they use movie reviews from Stanford Sentiment Treebank (SSTb) [23] and Stanford Twitter Sentiment corpus (STS) [20] containing 1.6 million tweets annotated with emoticons. Approach where emoticons or other syntactic units are used as an approximation to the true semantic annotation is called *distant supervision*.

In [24] authors propose a three step approach for sentiment classification. First, they train word embeddings on a large corpus of unlabeled tweets using a neural language model similar to C&W. Second, they use a convolutional neural network to further refine their embeddings on a corpus of 10 million tweets containing positive emoticons which are used for distant supervision, similar to the Stanford Twitter Sentiment corpus (STS) [20]. Finally, they use these embeddings and the parameters of their neural network to train the final model on the SemEval-2015 corpus of Tweets.

In [25] authors describe their Dynamical Convolutional Neural Network Architecture (DCNN) that uses dynamic k-max pooling. Difference from the standard max pooling is that k-max pooling returns subsequence of k maximum values in the sequence, instead of a single maximum value. They evaluate their model on several sentiment-related tasks. First one is the prediction of movie reviews in the Stanford Sentiment Treebank [23] consisting of close to ten thousand sentences labeled either *binary* (positive or negative) or *fine-grained* with five possible outcomes (negative, somewhat negative, neutral, somewhat positive, positive). It achieved accuracy of 86% and 48% on the binary and fine-grained task respectively. Second task is sentiment classification using Stanford Twitter Sentiment corpus (STS) [20]. They report accuracy of 87.4% which is higher than in other neural sentence models such as Max-TDNN [15] and Neural Bag-of-Words (NBoW).

III. Evaluation of distributed representation approaches for natural language processing

In this section I will present evaluation of C&W and other distributed representation models on sentiment analysis and other related tasks [26], [27]. Evaluations are performed either as a part of the related work in literature (section III-A) or as a competing entry in a machine learning challenge such as SemEval (section III-B).

A. Evaluation of word embeddings in literature

In [26] authors compare C&W model to the Turian's model [28], hierarchical log-bilinear model [29] and Huang's model [30] on several NLP tasks, including a two-class and a three-class sentiment polarity classification. For sentiment annotation they use Lydia's sentiment lexicon [31] which contains 6923 words.

Turian's model is essentially the same as C&W model, with the difference that it corrupts the last word in the n-gram instead of the middle one as in C&W model, and it uses separate learning rates for the embeddings and for the neural network weights [28].

Log-bilinear model [32] predicts embedding of a last word in n-gram using embeddings of all previous words through a linear model and log-bilinear loss function. Hierarchical version of a log-bilinear model [29] achieves efficiency by imposing a hierarchical structure on the words in the vocabulary (similar to [33]).

Huang's model [30] incorporates both local context on the level of a sentence and global context on the level of a document. Similar to the syntactic loss defined in C&W model (equation 4) and semantic loss in SSWE embedding (equation 5) they minimize loss between sentence s and sentence s^w where last word is replaced with w:

$$C_{s,d} = \sum_{w \in V} max(0, 1 - g(s,d) + g(s^w, d)) \qquad (7)$$

where g is a scoring function for sentence s and document d.

C&W model and Huang's model are comparable, with accuracy reaching over 85%, which is significantly better than other models whose accuracy was bellow 80%. C&W and Huang's model were also more accurate than other models on other classification tasks which included classification of noun gender, plurality and synonyms and antonyms.

Although embeddings provided by the models above are already very efficient, containing from 25 to 100 dimensions, authors investigate the impact of information reduction on each of the embeddings by either bitwise truncation (lowering the floating point precision for each of the dimensions) or principal component analysis (PCA). Surprisingly, even when each of the dimensions is truncated to just one bit of information (or, equivalently, by taking a sign of embedding value) the resulting accuracy drops by no more than 7%. PCA does not provide such an efficient reduction in information because removing all but a single component reduces accuracy by more than 15%, which is probably due to the inability of PCA to capture non-linear relationships in data.

In [27] authors describe their context-sensitive method based on neural networks for sentiment classification of tweets. They compare against NRC method (described in section

III-B) and SSWE method on their own dataset, on which they report slightly better results. They also compare with similar method that uses a context-based model [34] for sequential classification over streams of tweets.

B. Case study on Twitter sentiment classification challenge

The following section describes tasks of SemEval (Semantic Evaluation) challenges which relate to sentiment analysis. SemEval-2007 was the first challenge that featured semantic analysis as one of the tasks, and SemEval-2013 [5] was the first one that featured semantic analysis in Twitter.

Semeval-2013 task 2 [5] and Semeval-2014 task 9 [35] feature two subtasks: (i) *contextual polarity disambiguation*, where specific words of phrases were classified as positive, negative or neutral (given the message) and (ii) *message polarity classification*, where whole messages were classified as positive, negative or neutral. They also allowed teams to submit both *constrained* solutions, trained just on the given dataset, and *unconstrained* solutions, trained on any additional data which participants could collect. Semeval-2013 dataset consisted of Tweets and SMS messages, while Semeval-2014 added LiveJournal sentences along with more Tweets divided into two categories: regular and sarcastic.

Semeval-2015 task 10 [36] featured three additional subtasks, two of which related to the sentiment towards specific predefined topic in a tweet or a collection of Tweets, and the third whose goal was to determine the strengths of associations of terms with positive sentiment. The most recent SemEval challenge - SemEval-2016 Task 4 (http://alt.qcri.org/semeval2016/task4/) is a rerun of a SemEval-2015 Task 10 with an addition of several new subtasks that concentrate on finer qualification of sentiment. Specifically, instead of binary positive/negative classification the goal is to infer a percentage score, two-point, three-point and five-point scales of sentiment.

There are several observations that could be made regarding the most popular approaches. First, majority of approaches are supervised, using standard shallow classifiers like support vector machine (SVM), maximum entropy (MaxEnt) and Naive Bayes. Second, most important features are usually derived from some kind of sentiment lexicons, most popular being MPQA [37], and manual engineering of features using any domain-specific knowledge available.

The team that most successfully used this approach is NRC-Canada [38], who used their own set of manually engineered features called STATE:

- Presence of elongated words (for example, "coool").
- Numbers and categories of emoticons.
- Presence of punctuation sequences (for example, "?!" and "!!!").
- Usage of upper case.
- Usage of tokens which are categorized in sentiment lexicons (NRC Emotion Lexicon [39], [40], MPQA Lexicon [37] and Bing Liu Lexicon [41]).
- Usage of negation words.
- Usage of word ngrams and character ngrams.
- Position of a term.

Their machine learning methodology was rather standard - they used an SVM classifier with a linear kernel and cross-validation for regularization of a penalty parameter. They won SemEval-2013 and SemEval-2014 challenges, and some of the results they achieved were not surpassed even on SemEval-2015 challenge, on which NRC-Canada did not participate.

Starting from SemEval-2013 several teams used deep neural networks and word embeddings. One of them was the Coooolll system [18] that combined manually engineered STATE features along with the SSWE features in order to perform sentiment analysis of Tweets, using SVM with linear kernel as a learning algorithm. They ranked second on SemEval-2014. However, Coooolll system only leveraged word embeddings as an input features to an SVM classifier, which actually performed the classification.

In comparison, UNITN system [24] (described in section II-C) used deep convolutional neural network to both generate word embeddings and perform classification. It won several of the subtasks of SemEval-2015, beating even some of the results of NRC-Canada from the previous years. While it is probably too early to say that deep learning and word embedding approaches are becoming widely adopted, considering that majority of teams on SemEval-2015 used one of the standard supervised machine learning methods, it is promising to see these kind of systems finally achieving state-of-the-art performance on machine learning challenges.

IV. Discussion

SemEval challenges are a positive example how machine learning challenges could help identify state-of-the-art methods and promising approaches. In the recent years, at least in the field of sentiment analysis in Twitter, the SemEval challenges identified a shift towards more general methods such as neural network based models which learn directly from raw textual inputs, instead of leveraging manually engineered domain-specific features.

Are these approaches just a passing trend or a direction which will be more and more prevalent in the future natural language research? If we consider other machine learning challenges, for example ImageNet Large Scale Visual Recognition Challenge [42] which is held since 2010, then we can notice that they are already dominated by the deep learning approaches. General object recognition is a very hard task where manual engineering of features is not easy, and so the machine vision community was very eager to take advantage of deep learning methods to construct discriminative features automatically. In cases where objects being recognized come from a well specified domain, for example fingerprints or faces, there was no need to learn features automatically because many discriminative features already existed, and they easily achieved state-of-the-art performance. Similarly, I expect that deep learning methods will prove more and more useful in natural language processing as the challenges such as SemEval become more demanding, reflecting the rising expectations of the community.

ACKNOWLEDGMENT

This review paper has been written for the course: "Natural Language Processing" in the first year of postgraduate study 2014./2015. at the Faculty of Electrical Engineering and Computing, University of Zagreb, Croatia.

REFERENCES

[1] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?: Sentiment classification using machine learning techniques," in *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10*, ser. EMNLP '02. Association for Computational Linguistics, 2002, pp. 79–86.

[2] P. D. Turney, "Thumbs up or thumbs down?: Semantic orientation applied to unsupervised classification of reviews," in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ser. ACL '02. Association for Computational Linguistics, 2002, pp. 417–424.

[3] L. Jiang, M. Yu, M. Zhou, X. Liu, and T. Zhao, "Target-dependent twitter sentiment classification," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, ser. HLT '11. Association for Computational Linguistics, 2011, pp. 151–160.

[4] X. Hu, J. Tang, H. Gao, and H. Liu, "Unsupervised sentiment analysis with emotional signals," in *Proceedings of the 22Nd International Conference on World Wide Web*, ser. WWW '13. International World Wide Web Conferences Steering Committee, 2013, pp. 607–618.

[5] Z. K. V. S. A. R. Preslav Nakov, Sara Rosenthal and T. Wilson, "Semeval-2013 task 2: Sentiment analysis in twitter," in *Proceedings of the International Workshop on Semantic Evaluation*, ser. SemEval 2013, 2013, pp. 312—-320.

[6] P. D. Turney and P. Pantel, "From frequency to meaning: Vector space models of semantics," *Journal of Artificial Intelligence Research*, vol. 37, no. 1, pp. 141–188, 2010.

[7] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American Society for Information Science*, vol. 41, no. 6, pp. 391–407, 1990.

[8] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, ser. HLT '11, 2011, pp. 142–150.

[9] D. M. Blei, A. Y. Ng, M. I. Jordan, and J. Lafferty, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, p. 2003, 2003.

[10] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, "A neural probabilistic language model," *Journal of Machine Learning Research*, vol. 3, pp. 1137–1155, 2003.

[11] L. Y., B. Y., and H. G., "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[12] G. C. Tomas Mikolov, Kai Chen and J. Dean, "Efficient estimation of word representations in vector space," in *Proceedings of Workshop at International Conference on Learning Representations*, ser. ICLR '13, 2013.

[13] K. C. G. C. Tomas Mikolov, Ilya Sutskever and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proceedings of Conference on Neural Information Processing Systems*, ser. NIPS '13, 2013.

[14] Y. Bengio, "Neural net language models," *Scholarpedia*, vol. 3, no. 1, p. 3881, 2008.

[15] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proceedings of the 25th International Conference on Machine Learning*, ser. ICML '08, 2008, pp. 160–167.

[16] C. R., W. J., B. L., K. M., K. K., and K. P., "Natural language processing (almost) from scratch," *Journal of Machine Learning Research*, vol. 12, pp. 2493–2537, 2011.

[17] T. D., W. F., Y. N., Z. M., L. T., and Q. B., "Learning sentiment-specific word embedding for twitter sentiment classification," vol. 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*, ser. SemEval 2014, 2014, pp. 208–212.

[19] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, pp. 2121–2159, 2011.

[20] A. Go, R. Bhayani, and L. Huang, "Twitter sentiment classification using distant supervision," *Processing*, pp. 1–6, 2009.

[21] P. K. Novak, J. Smailovic, B. Sluban, and I. Mozetic, "Sentiment of emojis," *CoRR*, vol. abs/1509.07761, 2015. [Online]. Available: http://arxiv.org/abs/1509.07761

[22] C. N. dos Santos and M. Gatti, "Deep convolutional neural networks for sentiment analysis of short texts," in *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, ser. COLING 2014. Association for Computational Linguistics, 2014, pp. 69–78.

[23] J. W. J. C. C. D. M. A. Y. N. Richard Socher, Alex Perelygin and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2013, p. 1631–1642.

[24] A. Severyn and A. Moschitti, "Twitter sentiment analysis with deep convolutional neural networks," in *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '15. ACM, 2015, pp. 959–962.

[25] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," vol. 1, 2014, pp. 655–665.

[26] Y. Chen, B. Perozzi, R. Al-Rfou, and S. Skiena, "The expressive power of word embeddings," *CoRR*, vol. abs/1301.3226, 2013. [Online]. Available: http://arxiv.org/abs/1301.3226

[27] Y. Ren, Y. Zhang, M. Zhang, and D. Ji, "Context-sensitive twitter sentiment classification using neural network," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, ser. AAAI-16, 2016.

[28] J. Turian, L. Ratinov, and Y. Bengio, "Word representations: A simple and general method for semi-supervised learning," 2010, pp. 384–394.

[29] A. Mnih and G. Hinton, "A scalable hierarchical distributed language model," 2009, pp. 1081–1088.

[30] E. H. Huang, R. Socher, C. D. Manning, and A. Y. Ng, "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*, ser. ACL '12. Association for Computational Linguistics, 2012, pp. 873–882.

[31] N. Godbole, M. Srinivasaiah, and S. Skiena, "Large-scale sentiment analysis for news and blogs," in *Proceedings of the International Conference of Weblogs and Social Media (ICWSM*, 2007.

[32] A. Mnih and G. Hinton, "Three new graphical models for statistical language modelling," in *Proceedings of the 24th International Conference on Machine Learning*, ser. ICML '07. ACM, 2007, pp. 641–648.

[33] F. Morin and Y. Bengio, "Hierarchical probabilistic neural network language model," in *AISTATS'05*, 2005, pp. 246–252.

[34] A. Vanzo, D. Croce, and R. Basili, "A context-based model for sentiment analysis in twitter," in *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, ser. COLING 2014. Association for Computational Linguistics, 2014, pp. 2345–2354.

[35] P. N. Sara Rosenthal, Alan Ritter and V. Stoyanov, "Semeval-2014 task 9: Sentiment analysis in twitter," in *Proceedings of the International Workshop on Semantic Evaluation*, ser. SemEval 2014, 2014, pp. 73–80.

[36] S. K. S. M. M. A. R. Sara Rosenthal, Preslav Nakov and V. Stoyanov, "Semeval-2015 task 10: Sentiment analysis in twitter," in *Proceedings of the International Workshop on Semantic Evaluation*, ser. SemEval 2015, 2015, pp. 451–463.

[37] T. Wilson, J. Wiebe, and P. Hoffmann, "Recognizing contextual polarity in phrase-level sentiment analysis," in *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, ser. HLT '05. Association for Computational Linguistics, 2005, pp. 347–354.

[38] S. K. Saif Mohammad and X. Zhu, "Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets," in *Proceedings of the Seventh international workshop on Semantic Evaluation Exercises*, ser. SemEval 2013, 2013, pp. 321—-327.

[39] S. M. Mohammad and P. D. Turney, "Emotions evoked by common words and phrases: Using mechanical turk to create an emotion lexicon," in *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, ser. CAAGET '10. Association for Computational Linguistics, 2010, pp. 26–34.

[40] S. M. Mohammad and T. W. Yang, "Tracking sentiment in mail: How genders differ on emotional axes," in *Proceedings of the 2Nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis*, ser. WASSA '11. Association for Computational Linguistics, 2011, pp. 70–79.

[41] M. Hu and B. Liu, "Mining and summarizing customer reviews," in *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '04. ACM, 2004, pp. 168–177.

[42] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.