Tweester at SemEval-2017 Task 4: Fusion of Semantic-Affective and pairwise classification models for sentiment analysis in Twitter

Athanasia Kolovou [2,3], **Filippos Kokkinos** [1], **Aris Fergadis** [1,3], **Pinelopi Papalampidi** [1]
Elias Iosif [1,3], **Nikolaos Malandrakis** [4], **Elisavet Palogiannidi** [1], **Harris Papageorgiou** [3]
Shrikanth Narayanan [4], **Alexandros Potamianos** [1,3]

[1]School of ECE, National Technical University of Athens, Zografou 15780, Athens, Greece
[2] Department of Informatics, University of Athens, Athens, Greece
[3] "Athena" Research and Innovation Center, Maroussi 15125, Athens, Greece
[4] Signal Analysis and Interpretation Laboratory (SAIL), USC, Los Angeles, CA 90089, USA
akolovou@di.uoa.gr, el11142@mail.ntua.gr
fergadis@central.ntua.gr, el12003@central.ntua.gr
iosife@central.ntua.gr, malandra@usc.edu
epalogiannidi@gmail.com, xaris@ilsp.athena-innovation.gr
shri@sipi.usc.edu, potam@central.ntua.gr

Abstract

In this paper, we describe our submission to SemEval2017 Task 4: Sentiment Analysis in Twitter. Specifically the proposed system participated both to tweet polarity classification (two-, three- and five class) and tweet quantification (two and five-class) tasks. The submitted system is based on "Tweester" (Palogiannidi et al., 2016) that participated in last year's Sentiment analysis in Twitter Tasks A and B. Specifically it comprises of multiple independent models such as neural networks, semantic-affective models and affective models inspired by topic modeling that are combined in a late fusion scheme.

1 Introduction

Tweets are short length pieces of text, usually written in informal style that contain abbreviations, misspellings and creative syntax (like emoticons, hashtags etc). The challenging nature of sentiment analysis in Twitter motivated the organization of numerous tasks within the Semantics Evaluation (SemEval) workshop. In this paper, we show how our sentiment analysis framework called "Tweester" (winner of Subtask B in SemEval-2016 (Palogiannidi et al., 2016)), can be applied to all subtasks, namely Subtask A (message polarity classification), Subtask B (tweet classification according to a two-point scale), Subtask C (sentiment conveyed by a tweet towards the topic on a five-point scale), Subtask D (estimate the distribution of the tweets across a two-point scale), Subtask E (estimate the distribution of the tweets across a five-point scale). The system achieved high performance ranking 5th, 3rd, 4th, 5th and 8th for Subtasks A, B, C, D, and E, respectively (S.Rosenthal et al., 2017). In Section 2 we provide more details on the individual systems as well as the fusion scheme. Experimental procedure is described in Section 3 and some concluding remarks as well as an outlook on future work are presented in Section 4.

2 System Description

The submitted system is based on the fusion of several systems. Specifically the system consists of: 1) the semantic-affective system (submitted to the SemEval 2016 Task 4 (Palogiannidi et al., 2016)) that incorporates affective, semantic-similarity, sarcasm/irony and topic modeling features, 2) a single and a two-step convolutional neural network, 3) a system based on word embeddings, 4) a "stacking" based system that transforms the 3-class polarity problem of Subtask A, into 2-class binary problems and finally 5) the open-source system submitted to the SemEval 2015 Task 10 (Rosenthal et al., 2015a).

2.1 Preprocessing

We hypothesize that **hashtags** are able to express user's sentiment with regard to some topic or events (e.g., "Jazz all day #lovemusic"). Following this assumption, hashtag expansion into word strings (Palogiannidi et al., 2016) was performed using the Viterbi algorithm (Forney, 1973). The absolute and relative frequencies of hashtags to be expanded are used as features, as well as the binary indicators that a tweet contains hashtags that

Proceedings of the 11th International Workshop on Semantic Evaluations (SemEval-2017), pages 675–682,
Vancouver, Canada, August 3 - 4, 2017.

require expansion. **POS-tagging / Tokenization** is performed using the ARK NLP tweeter tagger (Owoputi et al., 2013). The Gensim model (Řehůřek and Sojka, 2010) is used which can automatically detect common **multi-word expressions (MWE)** from a stream of sentences. After we detect MWE, we select only the ones that consist up to two words, which we treat as a single token. **Negations** are usually expressions that are used to alter the sentiment orientations. We claim that not all parts of the tweet convey equally important information and some parts, like negated parts or the last words of a tweet may express an opposite meaning of what is literally said. The underlying intuition here is the cognitive dissonance phenomenon that is associated with the existence of ironic content, sarcasm and humour (Reyes and Rosso, 2014). Based on this claim, we detect the negation part of a tweet using the list proposed by (Potts, 2011). Specifically, when a negation token is detected, the tokens that follow are marked as negated until a punctuation mark is reached. Then, we extract features in the negated part. We also apply **context windows** on each tweet, in order to keep selected words. "Prefix" context windows are the first two and three tokens of the tweet, however , "suffix" windows change analogously to the length of the tweet, selecting the 20%, 50% and 70% of the last tokens.

2.2 Semantic-Affective system

The Semantic-Affective based system is the core model of "Tweester" which is based on previous work by (Malandrakis et al., 2014). The majority of the features used are affective ratings that have been estimated by semantic affective models, however, numerous non-affective or semantic features are also used.

2.2.1 Affective lexica

Using the semantic affective model described in (1) we created affective lexica by estimating continuous (in [-1,1]) ratings for unknown words. This model that was first proposed by (Malandrakis et al., 2013) and enhanced by (Palogiannidi et al., 2015) relies on the assumption that "semantic similarity implies affective similarity". First, a semantic model is built and then affective ratings are estimated for unknown tokens. This approach uses a set of words with known affective ratings, usually referred as seed words, as a starting point. The English manual annotated affective lexicon ANEW (Bradley and Lang, 1999) is used for selecting the seed words. The model is applicable both to single words or multi-word expression tokens:

$$\hat{v}(w) = \alpha_0 + \sum_{i=1}^{M} \alpha_i v(t_i) S(t_i, w), \qquad (1)$$

where $t_1 ... t_M$ are the seed words, $v(t_i)$ is the affective rating for seed word t_i, α_i is a trainable weight corresponding to seed t_i and $S(\cdot)$ is the semantic similarity metric between two tokens. The semantic model is built as shown in (Palogiannidi et al., 2015) using word-level contextual feature vectors and adopting a scheme based on mutual information for feature weighting. From the affective ratings we retain only the polarity features (instead of using additional affective dimensions, namely arousal and dominance). Affective lexica were created using a Twitter corpus, which we call *task-dependent corpus* and a generic corpus, which we call *out-of-domain* (see Section 3.1). In an attempt to create task-dependent affective lexica we use the out-of-domain corpus and follow a domain adaptation technique. Specifically, we build a language model using domain relevant sentences, i.e., tweets. Then, we estimate the perplexity of each out-of-domain sentence in order to evaluate its relevance to the language model. In this context, instances that are lexically more similar to the instances in the task-dependent corpus will be assigned lower perplexity scores. We create four adapted lexica selecting from the out-of-domain corpus the top 10%, 30%, 50% and 70% of the most relevant sentences to the language model.

Third party affective lexica are also used. Those include AFFIN ((Nielsen, 2011), NRC and nrctag (Mohammad et al., 2013)). Given the affective ratings, the next step is combining them through statistics. We use simple statistics grouped by part of speech tags. Specifically we compute: length (cardinality), min, max, max amplitude, sum, average, range (max minus min), standard deviation and variance. The results are statistics like "maximum valence among adjectives", "mean valence among proper nouns", "number of verbs and nouns", etc. All features mentioned above are not only extracted on the token-level, but also on the prefix and suffix parts of each tweet and the MWEs.

2.2.2 Additional features

In addition to the affective features, we also incorporate morphology, character and word embedding based features.

Character features include the frequencies of selected characters like capitalized letters, punctuation marks, emoticons and character repetition.

Word embeddings are utilized for the semantic similarity estimation. They were derived using word2vec (Mikolov et al., 2013b), representing each word as a d-dimensional vector. For each tweet the corresponding vectors of its constituent words are averaged to get a sentence-level feature vector.

As **subjectivity features** we use the absolute and the relative frequencies of the strong positive/negative and weak positive/negative words taken from a subjectivity lexicon (Wilson et al., 2005).

The detection of **irony** in tweets is mainly constituted from the detection of disagreement between what someone says and what he actually believes. According to this assumption, features that measure the level of opposition between literal and intended meanings in a tweet are extracted. Similar to (Barbieri and Saggion, 2014), we extracted the following features: i) *frequency based* which are features that are derived from the mean frequency of common and rare words in a tweet, as well as the difference between them. Word frequencies are indicated in the ANC Frequency Data corpus (Ide and Macleod, 2001), ii) *written-spoken gap* where we calculate the difference between the number of words that are considered "formal" vs. the "informal" ones in each tweet. Those words are also identified in the ANC corpus, iii) *sentiment distances* for which we first apply a threshold that separates words into positive and negative, based on polarity ratings from the lexica we describe in Section 2.2.1. For each tweet we compute: total sentiment range (average positive minus average negative), positive range (max positive minus tweet average), negative range (max negative minus tweet average) and iv) following the approach proposed by (Barbieri and Saggion, 2014), we use the same feature selection process related to synonyms, since they may be high indicators of irony.

A **topic modeling** method (described in Section 2.6), provides additional features to this system.

2.3 Convolutional Neural Network

In our framework we propose the combination of two neural networks. Specifically, we develop a deep Convolutional Neural Network (CNN) and a two-step Convolutional Neural Network. The neural network architecture is inspired by sentence classification tasks (Severyn and Moschitti, 2015; Kalchbrenner et al., 2014; Kim, 2014). Each tweet is represented by a sentence matrix $\mathbf{D}$ that is created as follows. First, each word is represented as a d-dimensional vector using word2vec (Mikolov et al., 2013b), and then, the word vectors are concatenated as follows:

$$\mathbf{D} = W_1 \oplus W_2 \oplus W_3 \oplus \cdots \oplus W_n., D \in \mathbb{R}^{d \times n} \quad (2)$$

where $\oplus$ indicates the concatenation operation. Each column i of $\mathbf{D}$ is a vector $W \in \mathbb{R}^d$ that corresponds to the i^{th} word of the tweet. This way the sequence of the words in the tweet is kept. In order to preserve the same length for all tweets, zero padding is applied by concatenating zero word vectors until the length n of the longest tweet is reached. The size of $\mathbf{D}$ is $d \times n$, where d is the dimension of the word embedding and n is the maximum number of words.

The matrix $\mathbf{D}$ is the input to the network, where a convolution operation is performed between $\mathbf{D}$ and a filter $F \in \mathbb{R}^{d \times m}$ which is applied to a window of m words to produce a new feature. The result of the convolutional layer is a vector $c \in \mathbb{R}^{n-h+1}$ (Kim, 2014). The network uses multiple m filters, with varying sliding windows and generates multiple features that are aggregated into a feature matrix $C \in IR^{mx(n-h+1)}$. The filters are learned during the training phase of the neural network (the exact parameter values are presented in Table 3). These features are the inputs to the next layer which selects the maximum value of each feature by applying a max-over-time pooling operation (max-pooling layer) (Collobert et al., 2011). Max pooling reduces the dimensionality of the input feature matrix and allows the "strongest" information to be considered in the resulting feature representation. The output pooled feature map matrix of this step has the form: $C_{pooled} \in \mathbb{R}^{\mathrm{m} \times \frac{\mathrm{n-h+1}}{\mathrm{s}}}$ (Kalchbrenner et al., 2014) where s is the length of each region. The next layer is a fully connected hidden layer that computes the following transformation: $\alpha(W_{hidden} * x + b_{hidden})$ as explained in (Nair and Hinton, 2010) where α is the rectified

linear activation function $relu(x) = max(0, x)$, $W_{hidden} \in \mathbb{R}^{m \times m}$ is the weight matrix and $b_{hidden} \in R^m$ is the bias. The output vector of this layer, $x \in R^m$ are the sentence embeddings for each tweet. Finally we add a soft-max layer that classifies the outputs of the hidden layer $x \in R^m$ to one of the possible classes.

Two-step CNN: In the case of 3-class problem of Subtask A we propose an additional two-step system. This process requires the re-annotation of the train datasets as follows. We separate the neutral tweets, while positive and negative tweets are annotated as "emotional". Then, we apply the aforementioned CNN model architecture which is trained on the re-annotated data. The output is predictions on neutral and "emotional" tweets. The next step involves the classification of the tweets that were found to belong to "emotional" category, into positive and negative. This step requires only the "emotional" tweets for training the CNN model.

2.4 Word2vec

This system uses word embeddings to predict the sentiment of each tweet in a supervised approach, using a classifier which is trained with the available labeled data. The vector for each word is a semantic description of how that word is used in context, so words that are used similarly in text will get similar vector representations. Motivated by this, we build this separate system that relies exclusively on tweets' semantic representation. Specifically the word embeddings of each tweet word are first extracted. Then, the vectors of each tweets' constituent words are averaged and form utterance-level vectors used for training the classifier.

2.5 Stacking

The main idea of this technique is to reduce a multi-class problem into binary 2-class problems and train one separate classifier for each pair of classes (Savicky and Fürnkranz, 2003). In the second step, the predictions of the binary classifiers are combined using a separate classifier. This process is referred to as stacking (Fürnkranz, 2001).

2.6 Topic modeling

Here, we perform sentence-level adaptation from the semantic space to the affective space. For the adaptation of the semantic space of each tweet we split a large Twitter corpus (see Section 3.1) in topics using LDA (Blei et al., 2003). We create a number of topics and an equal number of clusters with the following procedure. For each tweet we get the LDA posteriors which give the probabilities by which the tweet belongs to certain topics. The tweet is assigned to those clusters if the probability is above a threshold. Each cluster constitutes a sub-corpus for which a semantic model is built using word embeddings as features. The purpose of those steps is to calculate a new similarity score between a lexical token t_j and a seed word w_i using a semantic mixture of the above mentioned models as follows:

$$S(t_j, w_i) = \frac{\sum_{n=1}^{T} p(n|s) \cdot S_n(t_j, w_i)}{\sum_{n=1}^{T} p(n|s)}, \quad (3)$$

where $s = \{t_0, t_1, \ldots, t_j, \ldots, t_k\}$ are the tweet's tokens, w_i is the i_{th} seed word, T is the number of topics-clusters, $p(n|s)$ is the posterior probability for s to contain topic n and $S_n(\cdot)$ is the cosine similarity between t_j and w_i, obtained from cluster n. The similarities computed in (3) are used in (1). This enables the computation of affective scores for tweet tokens based on which the following statistics are computed: *max, min, mean, variance, standard deviation, range* (max - min), *extremum* (larger absolute value) and *sum*. We also compute the same statistics for the following POS tags of each tweet, *N, O, S, ^, Z, L, V, A, R, !*, using the ARK NLP tweeter tagger (Owoputi et al., 2013) getting max score over all nouns, min score over all nouns etc. Those values are normalized by dividing with the corresponding score computed over all tokens, e.g. the min score over all nouns is divided by the min score over all tokens.

2.7 Webis

We also incorporated the Webis system (Büchner and Stein, 2015), which is the ensemble of different subsystems (namely NRC, GUMLT, KlUE, TeamX) that ranked at the top of SemEval 2013 and 2014 Sentiment Analysis tasks (Nakov et al., 2013; Rosenthal et al., 2015b)

2.8 Fusion of systems

The last step of the "Tweester" framework is the combination of all the aforementioned systems that have been trained on different feature spaces. Specifically, this step applies late fusion by averaging the output posterior probabilities from each classifier.

3 Experimental procedure and results

3.1 Data

We train our systems using both general purpose and Twitter data. The training set is composed by the training, development and development-time testing data of SemEval-2013 and SemEval-2016, as described in Table 1. We also add to the train set, the test data from SemEval-2015 and SemEval-2014. We omit the SemEval-2016 test data, which are kept for testing and experimenting with our models. For the procedure of adaptive lexica creation we used a general purpose corpus that contains 116M sentences that was created by posing queries on a web search engine and aggregating the resulting snippets of web documents(Iosif et al., 2016). In addition, a Twitter-specific dataset is created and consists of 300M tweets (T-300M). Finally the ANEW lexicon (Bradley and Lang, 1999) is used for selecting the initial set of seed words of (1).

	Training Set
Subtask A	28,061
Subtask B & D	6,680
Subtask C & E	9,070

Table 1: Number of tweets used for training.

3.2 Systems

The **Semantic-Affective system** (see Section 2.2) is trained using the SemEval datasets of Table 1 for each subtask. We perform feature selection on the massive set of candidate features. Specifically, we perform a forward stepwise feature selection using a correlation criterion (Hall, 1999) that extracts the most informative features. For classification, a Naive Bayes tree classifier is trained. Naive Bayes trees proved superior to other types of trees during our testing, presumably due to the smoothing of observation distributions. This model is used for Subtasks A,B and D combined with the other systems, however in Subtask C and E it is used as a standalone system.

For the **word2vec-based system** (see Section 2.4) we trained a Random Forest classifier with 100 trees using the tweet-level vectors described in Section 2.4. The word embeddings are initialized using word2vec (Mikolov et al., 2013a,b) and are trained using the T-300M corpus (see Section 3.1). We apply a skip-gram model of window size 5 while the words with frequency less than 50 were not taken into consideration. The dimensionality of the word vectors used is $d = 50$. Words that appear in the tweet but do not have a vector representation, are initialized randomly from a uniform distribution.

The **stacking based system** (see Section 2.5) is used in Subtask A and requires that the training data is split into subsets using only examples from each of the two classes (i.e, positive-negative, positive-neutral and negative-neutral). We form tweet-level vectors (set to d=50), as in the word2vec based system, for each of the aforementioned subsets. Then, we train separate Random Forest classifiers with 100 trees, using the tweet-level vectors. After the training phase, each classifier is tested not only on the provided test data but also on the training data. The posterior probabilities from this step constitute the features for the classifier in the final step, which is a nearest neighbor classifier (Savicky and Fürnkranz, 2003).

We run a series of experiments with the **topic modeling system** (see Section 2.6) in which we fine-tune the following parameters. *Word2vec parameters* of topic clusters which are, size of word vectors, max skip length between words and the model's architecture, i.e. Continuous Bag-of-Words (CBOW) or skip-gram, the *number of topics* to extract from the T-300M, the *probability threshold* for grouping tweets into clusters (as described in Section 2.6) and the *number of seed words* to use from the ANEW lexicon in order to estimate the affective scores, as described in (1) and (3). Each experiment produces a different feature set. In order to select the best set for the semantic-affective based system, we evaluate them against the SemEval labeled data using a Naive Bayes tree classifier. The feature set that gives the best performance is selected for each subtask.

	Parameters
Number of convolutional filters	m = 200
Filter window size h	[3,4,5]
Size of max-pooling interval	width = 6, s = 2
Activation function	relu
Adadelta parameters	$\epsilon = 10^{-6}$ and $\rho = 0.95$

Table 3: Summary of CNN parameters used.

For the **CNN model** we first run a distant-

			Tweester Systems								
Subtask	Perf.	Rank	NRC	GUMLT	KIUE	TeamX	Sem-Affect	CNN	2step CNN	Word2vec	Stack
A	0.659	5	0.617	0.613	0.593	0.615	0.606	0.621	0.613	0.593	0.575
B	0.854	3	×	0.752	×	×	0.843	0.851	×	0.791	×
C	0.623	4	×	×	×	×	0.623	×	×	×	×
D	0.057	5	×	0.093	×	×	0.062	0.052	×	0.079	×
E	0.365	8	×	×	×	×	0.365	×	×	×	×

Table 2: Individual system combinations and their performance.

supervised phase where we use emoticons to infer the polarity of a balanced set of 15M tweets. The word-embeddings, $D \in \mathbb{R}^{d \times n}$ are updated during both the distant and the supervised training phases, as back-propagation is applied through the entire network. The neural network is trained on the 15M tweets for one epoch, followed by a supervised training phase using SemEval labeled data. The dimensionality of the vector representation in the sentence matrix is set to $d = 50$. The same parameters are used in both single and 2-step CNN models. The CNN model is not used in Task C and D due to the lack of a large distant training dataset annotated in 5 classes. The network parameters are summarized in Table 3.

3.3 Results

In Table 2 the integrated systems' performances are depicted along with the submitted combination for each subtask (the omitted systems are denoted with ×). For Subtasks A and B the evaluation metric is macro-averaged recall (AvgR), for Subtask C it is the macro-averaged mean absolute error (MAE^M), for Subtask D the normalized cross-entropy (KLD) is used and for Subtask E the metric is called the Earth Mover's Distance (EMD). All the aforementioned metrics are defined in (Rosenthal et al., 2014).

For Subtask A we combined all individual systems and achieved an AvgR of 0.659. CNN proved to be the most robust individual system, achieving the highest performance (0.621) among the others. The two-step CNN achieved a slightly lower score compared to the single-step model. Since the CNN model is quite robust in distinguishing positive vs negative tweets it seems that the 2-step model makes more errors on the first step, which is the distinction between neutral and emotional class. For Subtasks B and D the stepwise based systems are omitted (since they involve binary classification). The selected combinations are based on our empirical results using SemEval-2016 test dataset. Particularly, in Subtask B, where we decided to omit three subsystems from Webis, the model was ranked at the 3rd place with 0.854 AvgR. Similarly, in Subtask D we omitted the same systems as in B and ranked in 5th place. The results for B and D show that the highest performance is achieved by the CNN followed by the semantic-affective system. However, in Subtask D the selected combination degraded the best performing system. Finally, for Subtasks C and E we submitted only the semenatic-affective based system, based on experiments conducted on SemEval-2016 test dataset.

4 Conclusions

We presented a system for the sentiment classification of tweets for the SemEval 2017 Task 4: Sentiment analysis in twitter. The system participated in Subtasks A, B, C, D and E and proved very successful, ranking on the top 5 places for the first four subtasks. Our framework is improved using a two-step CNN, a stacking-based approach for the 3-class problem and we incorporate new features using the adaptation of the semantic space to each tweet. Future work should focus on domain adaptation technique as we believe there is still room for improvement. Also, we aim to investigate in more depth the fusion of different systems.

Acknowledgments

This work has been partially funded by the BabyRobot project supported by the EU Horizon 2020 Programme, grant number 687831. We would like to thank Fenia Christopoulou for providing the implementation of the topic modeling system. Also, the authors would like to thank NVIDIA for supporting this work by donating a TitanX GPU and the members of the Computing Systems Laboratory (CSLab) of The National Technical University of Athens (NTUA) for their technical support.

References

F. Barbieri and H. Saggion. 2014. Automatic detection of irony and humour in twitter. In *Proceedings of International Conference on Computational Creativity*.

DM. Blei, A. Ng, and M. Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research* 3(Jan):993–1022.

M. Bradley and P. Lang. 1999. Affective norms for English words (ANEW): Instruction Manual and Affective Ratings. Technical report.

M. Büchner and B. Stein. 2015. Webis: An Ensemble for Twitter Sentiment Detection. *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)* page 582.

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

G.D Forney. 1973. The viterbi algorithm. *Proceedings of the IEEE* 61(3):268–278.

J. Fürnkranz. 2001. Round robin rule learning. In *Proceedings of the 18th International Conference on Machine Learning (ICML-01): 146–153*.

M.A Hall. 1999. *Correlation-based feature selection for machine learning*. Ph.D. thesis, The University of Waikato.

N. Ide and C. Macleod. 2001. The American National Corpus: A standardized resource for American English. In *Proceedings of Corpus Linguistics*. page 831836.

E. Iosif, S. Georgiladakis, and A. Potamianos. 2016. Cognitively Motivated Distributional Representations of Meaning. In *Proc. of the Language Resources and Evaluation Conference (LREC)*.

N. Kalchbrenner, E. Grefenstette, and P. Blunsom. 2014. A convolutional neural network for modelling sentences. *CoRR* abs/1404.2188.

Y. Kim. 2014. Convolutional neural networks for sentence classification. *CoRR* abs/1408.5882.

N. Malandrakis, M. Falcone, C. Vaz, J. Bisogni, A. Potamianos, and S. Narayanan. 2014. SAIL: Sentiment Analysis using Semantic Similarity and Contrast Features. In *Proc. of the 8th International Workshop on Semantic Evaluation (SemEval)*. pages 512–516.

N. Malandrakis, A. Potamianos, E. Iosif, and S. Narayanan. 2013. Distributional semantic models for affective text analysis. *IEEE Transactions on Audio, Speech and Language Processing* 21(11):2379–2392.

T. Mikolov, K. Chen, G. Corrado, and J. Dean. 2013a. Efficient estimation of word representations in vector space. *CoRR* abs/1301.3781.

T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Proc of. Advances in Neural Information Processing systems (NIPS)*. pages 3111–3119.

S. M. Mohammad, S. Kiritchenko, and X. Zhu. 2013. NRC-Canada: Building the State-of-the-Art in Sentiment Analysis of Tweets. In *Proc. of the 7th International Workshop on Semantic Evaluation (SemEval)*. pages 321–327.

V. Nair and GE. Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*. pages 807–814.

P. Nakov, Z. Kozareva, A. Ritter, S. Rosenthal, V. Stoyanov, and T. Wilson. 2013. Semeval-2013 task 2: Sentiment analysis in twitter. In *Proc. of the 10th International Workshop on Semantic Evaluation (SemEval)*.

F. Å. Nielsen. 2011. A new ANEW: Evaluation of a word list for sentiment analysis in microblogs. In *Proc. of the ESWC Workshop on Making Sense of Microposts*. pages 93–98.

O. Owoputi, B. O'Connor, C. Dyer, K. Gimpel, N. Schneider, and N. A Smith. 2013. Improved part-of-speech tagging for online conversational text with word clusters. In *In Proceedings of the North American Chapter of the Association for Computational Linguistics*.

E. Palogiannidi, E. Iosif, P. Koutsakis, and A. Potamianos. 2015. Valence, Arousal and Dominance Estimation for English, German, Greek, Portuguese and Spanish Lexica using Semantic Models. In *Proc. of Interspeech*. pages 1527–1531.

E. Palogiannidi, A. Kolovou, F. Christopoulou, E. Iosif, N. Malandrakis, H. Papageorgiou, S. Narayanan, and A. Potamianos. 2016. Tweester at SemEval 2016: Sentiment Analysis in Twitter using Semantic-Affective Model Adaptation. In *Proc. of the 10th International Workshop on Semantic Evaluation (SemEval)*. pages 155–163.

C. Potts. 2011. Sentiment symposium tutorial. In *Proc. of Sentiment Symposium Tutorial*.

R. Řehůřek and P. Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proc. of the Language Resources and Evaluation Conference (LREC) Workshop on New Challenges for NLP Frameworks*. pages 45–50.

A. Reyes and P. Rosso. 2014. On the difficulty of automatically detecting irony: beyond a simple case of negation. *Knowledge and Information Systems* 40(3):595–614.

S. Rosenthal, P. Nakov, S. Kiritchenko, S. M. Mohammad, A. Ritter, and V. Stoyanov. 2015a. Semeval-2015 Task 10: Sentiment Analysis in Twitter. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval)*. Association for Computational Linguistics (ACL), pages 451–463.

S. Rosenthal, P. Nakov, S. Kiritchenko, S. M Mohammad, A. Ritter, and V. Stoyanov. 2015b. Semeval-2015 task 10: Sentiment analysis in twitter. *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)* .

S. Rosenthal, A. Ritter, P. Nakov, and V. Stoyanov. 2014. SemEval-2014 task 9: Sentiment Analysis in Twitter. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval)*. Association for Computational Linguistics (ACL), pages 73–80.

P. Savicky and J. Fürnkranz. 2003. Combining pairwise classifiers with stacking. In *International Symposium on Intelligent Data Analysis*. Springer, pages 219–229.

A. Severyn and A. Moschitti. 2015. Unitn: Training deep convolutional neural network for twitter sentiment classification. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), Association for Computational Linguistics*. pages 464–469.

S.Rosenthal, N.Farra, and P.Nakov. 2017. SemEval-2017 task 4: Sentiment analysis in Twitter. In *Proceedings of the 11th International Workshop on Semantic Evaluation*. Association for Computational Linguistics, SemEval '17.

T. Wilson, J. Wiebe, and P. Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proc. of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT/EMNLP)*. pages 347–354.