SENSEI-LIF at SemEval-2016 Task 4: Polarity embedding fusion for robust sentiment analysis

Mickael Rouvier
Aix-Marseille Université
CNRS, LIF UMR 7279
13000, Marseille, France
mickael.rouvier@lif.univ-mrs.fr

Benoit Favre
Aix-Marseille Université
CNRS, LIF UMR 7279
13000, Marseille, France
benoit.favre@lif.univ-mrs.fr

Abstract

This paper describes the system developed at LIF for the SemEval-2016 evaluation campaign. The goal of Task 4.A was to identify sentiment polarity in tweets. The system extends the Convolutional Neural Networks (CNN) state of the art approach. We initialize the input representations with embeddings trained on different units: lexical, part-of-speech, and sentiment embeddings. Neural networks for each input space are trained separately, and then the representations extracted from their hidden layers are concatenated as input of a fusion neural network. The system ranked 2nd at SemEval-2016 and obtained an average F1 of 63.0%.

1 Introduction

This paper describes the system developed at LIF for the SemEval-2016 sentiment analysis evaluation task (Nakov et al., 2016). The goal of our participation was to apply approaches developed for the European FP7 project SENSEI [1] based on the study of human conversations according to feelings, opinions, emotions of the participants, in corpora such as transcripts of telephone speech and web comments.

We have participated in Subtask A: sentiment analysis at the message level. It consists in determining the message polarity of each tweet in the test set. The sentiment polarity classification task is set as a three-class problem: positive, negative and neutral.

The sentiment analysis task is often modeled as a classification problem which relies on features extracted from the text in order to feed a classifier. Recent work has shown that Convolutional Neural Networks (CNN) using word representations as input are well suited for sentence classification problems (Kim, 2014) and have been shown to produce state-of-the-art results for sentiment polarity classification (Tang et al., 2014a; Severyn and Moschitti, 2015). Pre-trained word embeddings are used to initialize the word representations, which are then taken as input of a text CNN.

Our approach consists in learning polarity classifiers for three types of embeddings, based on the same CNN architecture. Each set of word embedding models the tweet according to a different point of view: lexical, part-of-speech and sentiment. A final fusion step is applied, based on concatenating the hidden layers of the CNNs and training a deep neural network for the fusion.

Our contributions are as follows:

- We extend the deep CNN architecture proposed in (Poria et al., 2015) and introduce lexical information similar to (Ebert et al., 2015).
- We introduce **polarity embeddings**, tweet representations extracted from the hidden layer of CNNs with different word embeddings as input.
- We fuse polarity embeddings by concatenating them and feeding them to a neural network trained on the final task.
- The source code of our system, the models trained for the evaluation, and the corpus collected for creating word embeddings are made

[1] http://www.sensei-conversation.eu/

Proceedings of SemEval-2016, pages 202–208,
San Diego, California, June 16-17, 2016. ©2016 Association for Computational Linguistics

available to the community to help future research [2].

The paper is structured as follows. Section 2 presents the system architecture. Section 3 reviews the implementation details. Then we detail the different word embeddings and other features used in our system (Section 4). Results and discussion appear in Section 5.

2 Polarity embeddings

Deep learning models have been shown to produce state-of-the-art performance in various domains (vision, speech, etc...). Convolutional Neural Networks (CNN) represent one of the most used deep learning model in computer vision (LeCun and Bengio, 1995). Recent work has shown that CNNs are also well suited for sentence classification problems and can produce state-of-the-art results (Tang et al., 2014a; Severyn and Moschitti, 2015). The difference between CNNs applied to computer vision and their equivalent in NLP lies in the input dimensionality and format. In computer vision, inputs are usually single-channel (eg. grayscale) or multi-channel (eg. RGB) 2D or 3D matrices, usually of constant dimension. In sentence classification, each input consists of a sequence of words of variable length. Each word w is represented with a n-dimensional vector (word embedding) e_w of constant size. All the word representations are then concatenated in their respective order and padded with zero-vectors to a fixed length (maximum possible length of the sentence).

Word embeddings are an approach for distributional semantics which represents words as vectors of real numbers. Such representation has useful clustering properties, since it groups together words that are semantically and syntactically similar (Mikolov et al., 2013). For example, the word “coffee” and “tea” will be very close in the created space. The goal is to use these features as input to a CNN classifier. However, with the sentiment analysis task in mind, typical word embeddings extracted from lexical context might not be the most accurate because antonyms tend to be placed at the same location in the created space. As exemplified in Table 1, “good” and “bad” occur in similar contexts, and therefore obtain very similar representations. In addition, the model does not differentiate the senses of a word and creates a representation close to the most used sense in the training data. In order to tackle the representation robustness problem, we propose to extract word embeddings with different training regimes, and fuse their contribution to the system.

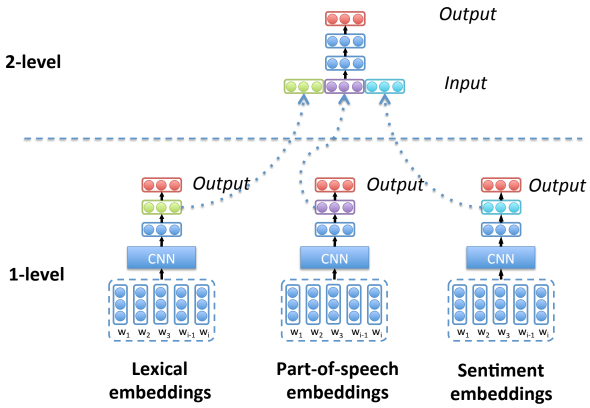
Output
2-level
Input
Output
Output
Output
CNN
CNN
CNN
1-level
w_1 w_2 w_3 w_{i-1} w_i
w_1 w_2 w_3 w_{i-1} w_i
w_1 w_2 w_3 w_{i-1} w_i
Lexical embeddings
Part-of-speech embeddings
Sentiment embeddings

Figure 1: Overview of the sentiment embedding fusion approach. In a first level, different word representations are used to train CNNs to predict sentiment polarity, in the second level, representations extracted at hidden layers are concatenated and fed to a final classifier.

There are two approaches commonly used for fusion: early and late fusion. Late fusion considers that the different systems are independent by first applying classification separately on each system and then merging the output using a high-level classifier. Unfortunately, the classifier cannot model the correlations among modalities. The early fusion approach tackles this problem by learning features and class relationships to model the interaction between modalities. While late fusion cannot benefit from different system feature correlations, early fusion requires lots of training data.

In previous work (Rouvier et al., 2015), we have proposed a new fusion framework called embedding fusion which consists in concatenating hidden layers of subsystems trained independently, and input them to an other classifier trained to the actual task targets. This embedding fusion approach goes beyond late fusion and overcomes most of the problems linked to early fusion.

In this paper, we apply the embedding fusion to

[2] http://www.github.com/mrouvier/SemEval2016

Lexical		Part-of-speech		Sentiment	
good	**bad**	**good**	**bad**	**good**	**bad**
great	good	great	good	great	terrible
bad	terrible	bad	terrible	goid	horrible
goid	baaad	nice	horrible	nice	shitty
gpod	horrible	gd	shitty	goood	crappy
gud	lousy	goid	crappy	gpod	sucky
decent	shitty	decent	baaaad	gd	lousy
agood	crappy	goos	lousy	fantastic	horrid
goood	sucky	grest	sucky	wonderful	stupid
terrible	horible	guid	fickle-minded	gud	:/
gr8	horrid	goo	baaaaad	bad	sucks

Table 1: Closest words to “good” and “bad” according to different regimes for creating word embeddings: lexical, part-of-speech and sentiment (described later in the paper).

the sentiment polarity prediction task, with a two-level architecture (Figure 1). Given a tweet, the first level extracts input representations based different word embeddings. These embeddings are fed to a CNN with n-gram filters (from 1 to 5). The CNN is followed by a series of fully connected hidden layers which are trained to predict the target (sentiment polarity). Three different sets of word embeddings are used: lexical embeddings, joint lexical-part-of-speech embeddings, and joint lexical-sentiment embeddings. The training procedure for the embeddings is explained in the following sections.

The second level inputs the concatenation of the last hidden layer resulting from each input representation, which we call **polarity embeddings**. This representation is fed to fully connected hidden layers, and also trained to predict the polarity target. This method allows us to take advantage of both early and late fusion at the same time, which brings an improvement in term of performance over merging the decisions of the independent neural networks.

3 Implementation details

The proposed architecture relies on word embeddings as word representation as well as sentiment polarity lexicon features, concatenated to the word representation. An alternative to word-level features captured by CNNs is to extract sentence-level features in order to model global evidence in the tweet. In order to incorporate this source of information into the system, a classical MLP with one hidden layer is trained to predict sentiment polarity from a set of sentence-level features and its hidden layer is concatenated to the other polarity embeddings and fed to the second-level MLP. The CNN and MLP are trained jointly.

The final complete architecture including CNNs and the sentence-level MLP, presented in Figure 2, is based on a single convolutional layer followed by a max-over-time pooling layer (Collobert et al., 2011) and two fully-connected layers. In order to learn this kind of model there are two soft-max fully connected layers. The first one is connected to the pooling layer and the second one at the end of fully-connected layer.

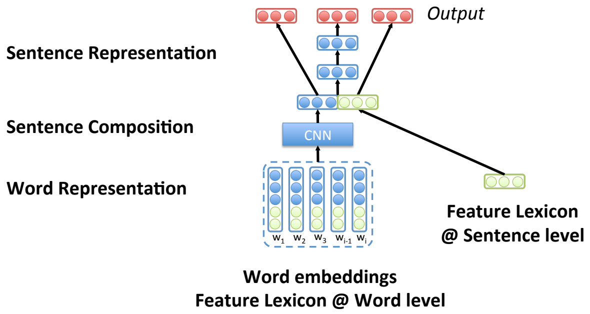

Figure 2: Actual CNN architecture: word representations are concatenated with lexicon features, and sentence-level lexicon features are concatenated with the polarity embeddings, and also trained to predict polarity targets on its own.

The parameters of our model were chosen so as to maximize performance on the development set: the width of the convolution filters is set to 5 and the number of convolutional feature maps is 500. We use ReLU activation functions and a simple max-pooling. The two fully connected hidden-layers are of size 512. For each layer, a standard dropout of 0.4 (40 % of the neurons are disabled in each iteration) is used. The back-propagation algorithm used for training is Adadelta. In our experiments we observed that the weight initialization of the convolution layer can lead to high variation in term of performance. Therefore, we trained 20 models and selected the one that obtained the best results on the development corpus.

In the second part of the system which inputs polarity embeddings and predicts polarity targets, the DNN is composed of two 512-dimensional hidden layers. The non-linearity of the hidden layers is corrected by a ReLU function.

4 Input features

4.1 Word embeddings

We propose to make use of word embeddings trained under different regimes, in order to capture different aspects of the relation between words so that it might benefit the polarity classifier. Three representations are explored.

Lexical embeddings: these embeddings are obtained with the classical skipgram model from (Mikolov et al., 2013). The representation is created by using the hidden layer of a linear neural network to predict a context window from a central word. For a given context $w_{i-2} \dots w_{i+2}$, the input to the model is w_i, and the output could be w_{i-2}, w_{i-1}, w_{i+1}, w_{i+2}. This method typically extracts a representation which both covers syntax and semantics, to some extent.

Part-of-speech embeddings: as stated earlier, the lexical model cannot distinguish between the senses of words and creates a single representation per word form. For example, the word "apple" receives an embedding that is a mixture of its different contextual senses: fruit, company... A lot of sophisticated approaches have been proposed to tackle the problem (Guo et al., 2014; Neelakantan et al., 2015; Huang et al., 2012), by considering senses as latent variables during training, or by conditionning the training documents on topic distributions. In our system we follow a very simple approach which creates joint embeddings for words and their part of speech. Thus, for context $w_{i-2} \dots w_{i+2}$ tagged with the part-of-speech sequence $p_{i-2} \dots p_{i+2}$ the input to the model is (w_i, p_i) and the output is (w_{i-2}, p_{i-2}), (w_{i-1}, p_{i-1}), $(w_{i+1} : p_{i+1})$, (w_{i+2}, p_{i+2}).

Sentiment embeddings: another problem with the basic skipgram approach (lexical embeddings) is that the model ignores the sentiment polarity of the words. As a result, words with opposite polarity, such as "good" and "bad", are mapped into close vectors. In (Tang et al., 2014b), the authors propose to tackle this problem so that sentiment information is encoded in the continuous representation of words. They propose to create a neural network that predicts two tasks: the context of the word and the sentiment label of the whole sentence. Since it is expensive to manually label sentences with a polarity label, the authors propose to use tweets that contain emoticons and rely on the polarity of the emoticon to label the sentences. As they report that best performance is obtained by weighting both tasks equivalently, the model is the same as for lexical embeddings, except that the predicted context is formed of (word, sentiment) couples. For example, if s is the polarity of the sentence where the context $w_{i-2} \dots w_{i+2}$ is extracted, the model gets w_i as input and has to predict $(w_{i-2}, s), (w_{i-1}, s), (w_{i+1}, s), (w_{i+2}, s)$.

4.2 Sentiment lexicon features

Word representations are learned from distributional information of words in large corpora. Although such statistics are semantically informative, they disregard the valuable information that is contained in manually curated sentiment lexicons. In (Ebert et al., 2015), the authors propose to incorporate knowledge from semantic lexicons at the word level. The goal is to extract features based on the overlap between words in the input and sentiment lexicons, and stack these features to the word embedding.

We create two such features per word per lexicon. Both are binary indicators of positive and negative polarity of that word in the lexicons. The lexicons for this feature type are MPQA (Wiebe et al., 2005), Opinion lexicon (Hu and Liu, 2004), and NRC Emotion lexicon (Mohammad and Turney, 2013). The NRC lexicons provide a score for each word instead of just a label. We replace the binary indicators by the scores.

4.3 Sentence-level features

The following features are extracted at sentence level and used for training the sentence-level MLP:

- **Lexicons**: frequency of lemmas that are matched in MPQA (Wiebe et al., 2005), Opinion Lexicon (Hu and Liu, 2004) and NRC Emotion lexicon (Mohammad and Turney, 2013).
- **Emoticons**: number of emoticons that are grouped in positive, negative and neutral categories.
- **All-caps**: number of words in all-caps
- **Elongated units**: number of words in which characters are repeated more than twice (for example: loooooool)
- **Punctuation**: number of contiguous sequences of period, exclamation mark and question mark.

5 Experiments

5.1 Pre-processing

A step of pre-processing is applied to every tweet in the corpus:

- **Character encoding:** every tweet is encoded in UTF-8
- **XML Entities:** all the XML entities are converted back to characters
- **Lowercase:** all the characters are converted in lowercase
- **Lengthening:** character lengthening consists in repeating several times a character in a word. It is used in social media as a method to emphasize a fact. This extension is often correlated with the expression of sentiment. If a character is repeated more than three times, we reduce it to three characters. For example, “looool” is replaced by “loool”.
- **Tokenization:** tokenization is performed by splitting a sentence in pre-lexical units. We used the tokenizer from the macaon toolchain (Nasr et al., 2011). It is based on a regular grammar that defines a set of types of atoms. A lexical analyzer detects the character sequences (in terms of the grammar) and combines them as a type. We added the atoms for detecting smileys, hashtags and users names (atoms specific to tweets).
- **Map generic words:** The hashtags, numbers and usertags are mapped to generic tokens.

5.2 Corpus

We use the train and dev corpora from Twitter’13 to 16 for training and Twitter’16-dev as a development set. Note that we were unable to download all the training and development data because some tweets were deleted or not available due to modified authorization status. The datasets are summarized in Table 3:

Corpus	Positive	Negative	Neutral	Total
Train	7.727	2.916	7.837	18.480
Dev	884	279	616	1.779

Table 3: Statistics of the successfully downloaded part of the SemEval 2016 Twitter sentiment classification dataset.

5.3 Word embedding training

To train the word embeddings, we have created a unannotated corpus of sentiment bearing tweets in English. These tweets were recovered on the Twitter platform by searching for emotion keywords (from the sentiment lexicons) and unigrams, bigrams and trigrams extracted from the SemEval training corpus. This corpus consists of about 90 million tweets. A sub-corpus of about 20 million tweets containing at least one emoticon is used for training the sentiment embeddings. Both corpora are made available [3].

In our experiments, lexical embeddings and part-of-speech embeddings are estimated using the word2vec toolkit (Mikolov et al., 2013). Sentiment embeddings are estimated using word2vecf. This toolkit allows to replace linear bag-of-word contexts with arbitrary features. The embeddings are trained using the skipgram approach with a window of size 3 and 5 iterations. The dimension of the embeddings is fixed to 100. Part-of-speech tagging is performed with Tweet NLP (Owoputi et al., 2013; Gimpel et al., 2011).

5.4 Results

Overall performance: The evaluation metric used in the competition is the macro-averaged F-measure calculated over the positive and negative categories. Table 4 presents the overall performance of our system. It achieved the second rank on the Twitter 2016

[3] http://www.github.com/mrouvier/SemEval2016

Feature set	Lexical	Part-of-speech	Sentiment	SENSEI-LIF
all features	61.3	62.0	62.3	63.0
w/o word level lexicon	61.7 (+0.4)	62.4 (+0.4)	61.6 **(-0.7)**	63.2 (+0.2)
w/o sentence level lexicon	60.7 **(-0.6)**	61.1 **(-0.9)**	62.0 **(-0.3)**	62.6 **(-0.4)**
w/o both lexicon	61.0 **(-0.3)**	61.4 **(-0.6)**	61.8 **(-0.5)**	62.8 **(-0.2)**
w/o word embeddings	58.4 **(-2.9)**	59.1 **(-2.9)**	59.6 **(-2.7)**	59.6 **(-3.4)**

Table 2: Ablation experiment: macro-averaged F-scores obtained on the Twitter 2016 test sets with each of the feature groups removed.

data among 34 teams. The system proved to generalize well to other types of short informal texts; it ranked first and third respectively on the two out-of-domain datasets: Live Journal 2014 and SMS 2013.

Corpus	SENSEI-LIF	Rank
Twt2013	70.6	3
SMS2013	63.4	3
Twt2014	74.4	1
TwtSarc2014	46.7	8
LvJn2014	74.1	1
Twt2015	66.2	2
Twt2016	63.0	2

Table 4: Overall performance of the SENSEI-LIF sentiment analysis systems.

Contribution of features: Table 2 presents the results of ablation experiments on the Twitter 2016 test set. *SENSEI-LIF* is the system which participated to the evaluation campaign. We present the results of three contrastive systems: *Lexical*, *Part-of-speech* and *Sentiment*. These systems are based on the CNN classifier prior to the concatenation of the hidden layers. They use only one set of word embeddings without any kind of fusion.

The different features used in our system are: lexicon features and word embeddings. The ablation of lexicon features removes the lexicon features at the word and sentence level. The ablation of word embeddings feature consists in randomly initializing the word representations.

We observe that the most influential features are word embeddings. They provide a gain of 3.4 points. The main advantage of word embeddings is to learn unsupervised representations on very large corpora which capture general semantic properties. The last most important features are lexicon features. We observe that word level lexicon features are not relevant and tend to degrade the performance of the *SENSEI-LIF* system on the Twitter 2016 dataset.

Impact of fusion: Table 5 presents the results using different kinds of fusion: early, late and embedding fusion. We observe that early fusion obtains the worse results. We think that is due to the small training corpus used. Embedding fusion obtains the bests results on the Twitter 2016 dataset, but more generally late and embedding fusions obtain very close results on the other datasets.

Corpus	Early	Late	Embedding
Twt2013	69.4	70.4	**70.6**
SMS2013	62.6	**63.7**	63.4
Twt2014	73.4	74.3	**74.4**
TwtSarc2014	46.2	44.7	**46.7**
LvJn2014	74.3	**74.4**	74.1
Twt2015	64.1	**66.8**	66.2
Twt2016	61.6	62.8	**63.0**

Table 5: Overall performance using different methods of fusion: early, late and embedding fusion.

6 Conclusions

This paper describes the LIF participation at SemEval 2016. Our approach consists in learning polarity classifiers for three types of embeddings, based on the same CNN architecture. Each set of word embeddings models the tweet according to a different point of view: lexical, part-of-speech and sentiment. A final fusion step is applied, based on concatenating the hidden layers of the CNNs and training a deep neural network for the fusion. The fusion system ranked 2nd at the SemEval-2016 evaluation campaign.

Acknowledgments

The research leading to these results has received funding from the European Union - Seventh Framework Programme (FP7/2007-2013) under grant agreement 610916 SENSEI [4]. The Tesla K40 used for this research was donated by the NVIDIA Corporation.

[4] http://www.sensei-conversation.eu

References

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *JMLR*, 12:2493–2537.

Sebastian Ebert, Ngoc Thang Vu, and Hinrich Schütze. 2015. A linguistically informed convolutional neural network. In *Computational Approaches to Subjectivity, Sentimant and Social Media Analysis WASSA*, page 109.

Kevin Gimpel, Nathan Schneider, Brendan O'Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah A Smith. 2011. Part-of-speech tagging for twitter: Annotation, features, and experiments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 42–47. Association for Computational Linguistics.

Jiang Guo, Wanxiang Che, Haifeng Wang, and Ting Liu. 2014. Learning sense-specific word embeddings by exploiting bilingual resources. In *COLING*, pages 497–507.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *SIGKDD*, pages 168–177.

Eric H Huang, Richard Socher, Christopher D Manning, and Andrew Y 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. Association for Computational Linguistics.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Yann LeCun and Yoshua Bengio. 1995. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10).

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Saif M Mohammad and Peter D Turney. 2013. Nrc emotion lexicon. Technical report, NRC Technical Report.

Preslav Nakov, Alan Ritter, Sara Rosenthal, Veselin Stoyanov, and Fabrizio Sebastiani. 2016. SemEval-2016 task 4: Sentiment analysis in Twitter. In *Proceedings of the 10th International Workshop on Semantic Evaluation*, SemEval '16, San Diego, California, June. Association for Computational Linguistics.

Alexis Nasr, Frédéric Béchet, Jean-François Rey, Benoît Favre, and Joseph Le Roux. 2011. Macaon: An nlp tool suite for processing word lattices. In *ACL*, pages 86–91.

Arvind Neelakantan, Jeevan Shankar, Alexandre Passos, and Andrew McCallum. 2015. Efficient non-parametric estimation of multiple embeddings per word in vector space. *arXiv preprint arXiv:1504.06654*.

Olutobi Owoputi, Brendan O'Connor, Chris Dyer, Kevin Gimpel, Nathan Schneider, and Noah A Smith. 2013. Improved part-of-speech tagging for online conversational text with word clusters. Association for Computational Linguistics.

Soujanya Poria, Erik Cambria, and Alexander Gelbukh. 2015. Deep convolutional neural network textual features and multiple kernel learning for utterance-level multimodal sentiment analysis. In *Proceedings of EMNLP*, pages 2539–2544.

Mickael Rouvier, Sebastien Delecraz, Benoit Favre, Meriem Bendris, and Frederic Bechet. 2015. Multimodal embedding fusion for robust speaker role recognition in video broadcast. In *ASRU*.

Aliaksei Severyn and Alessandro 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, Denver, Colorado*, pages 464–469.

Duyu Tang, Furu Wei, Bing Qin, Ting Liu, and Ming Zhou. 2014a. Coooolll: A deep learning system for twitter sentiment classification. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 208–212.

Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. 2014b. Learning sentiment-specific word embedding for twitter sentiment classification. In *ACL (1)*, pages 1555–1565.

Janyce Wiebe, Theresa Wilson, and Claire Cardie. 2005. Annotating expressions of opinions and emotions in language. *Language resources and evaluation*, 39(2-3):165–210.