IITP at EmoInt-2017: Measuring Intensity of Emotions using Sentence Embeddings and Optimized Features

Md Shad Akhtar *, **Palaash Sawant** +, **Asif Ekbal** *, **Jyoti Pawar** +, **Pushpak Bhattacharyya** *
* Indian Institute of Technology Patna, India
[shad.pcs15,asif,pb]@iitp.ac.in
+ Goa University, India
palaash77@gmail.com, jdp@unigoa.ac.in

Abstract

This paper describes the system that we submitted as part of our participation in the shared task on Emotion Intensity (EmoInt-2017). We propose a Long short term memory (LSTM) based architecture cascaded with Support Vector Regressor (SVR) for intensity prediction. We also employ Particle Swarm Optimization (PSO) based feature selection algorithm for obtaining an optimized feature set for training and evaluation. System evaluation shows interesting results on the four emotion datasets i.e. *anger*, *fear*, *joy* and *sadness*. In comparison to the other participating teams our system was ranked 5th in the competition.

1 Introduction

Emotion analysis (Picard, 1997) deals with automatic extraction of emotion expressed in a user written text. Basic emotions expressed by a human being, as categorized by Ekman (1992), are *joy*, *sadness*, *surprise*, *fear*, *disgust* and *anger*. With the growing amount of social media generated text it has become a challenging task to efficiently mine emotions of the user. However, finding only the emotion does not always reflect exact state of mood of a user. Level or intensity of emotion often differs on a case-to-case basis within a single emotion. Some emotions are gentle (e.g '*not good*') while others can be very severe (e.g. '*terrible*'). Finding the intensity level of the expressed emotion is another non-trivial task that researchers have to face.

The shared task on Emotion Intensity (EmoInt-2015) (Mohammad and Bravo-Marquez, 2017) was targeted to build an efficient system for intensity prediction on a continuous scale of 0 (least intense) to +1 (most intense). There were four datasets collected from Twitter, each reflecting one class of emotion i.e. *anger*, *fear*, *joy* and *sadness*, respectively.

We propose a Long Short Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) based neural network architecture cascaded with Support Vector Regression (SVR) (Smola and Schölkopf, 2004). We build our system on top of word embeddings along with the assistance of an optimized feature set obtained through Particle Swarm Optimization (PSO) (Kennedy and Eberhart, 1995). A major hurdle in obtaining a good word representation was the noisy and informal nature of text. Therefore, in the preliminary step, we perform a series of normalization heuristics in line with (Akhtar et al., 2015). The word embeddings of the resultant normalized text was more representative than that of the unnormalized text.

The high-dimensionality of feature vector often contributes to high complexity of the system. Also, some features have high degree of relevance towards a particular task/domain than the others. Careful selection of features for any task often leads to improved system performance. However, finding the relevant set of features is cumbersome and time-consuming task. Motivated by this we employ a Particle Swarm Optimization (PSO) based feature selection technique for selecting a subset of features from a feature pool. By utilizing the reduced and pruned feature set for training and evaluation, resultant system often performs considerably well. At the same time complexity of the system also reduces as it requires fewer parameters to learn. Literature survey shows successful application of PSO for various tasks and/or domains (Lin et al., 2008; Akhtar et al., 2017; Yadav et al., 2017).

Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, pages 212–218
Copenhagen, Denmark, September 7–11, 2017. ©2017 Association for Computational Linguistics

2 System Description

This section discusses our proposed approach in detail. The subsequent subsections present various components of our system.

2.1 Pre-processing and Normalization

- **Mentions, URLs and Punctuations:** In this step we filter out all the user mentions and URLs as they do not have any emotional bondings. Secondly, we strip off all the punctuations from the word boundaries to make it a valid dictionary word, e.g. 'first//' to 'first'. Improper use of punctuation was one of the reasons for data sparsity, when working with distributed word representation. After employing this step we observed that the number of out-of-vocabulary (OOV) words are effectively reduced.

- **Hashtag Segmentation:** Here the '#' symbol is stripped off from the hashtags. The resulting token is split into constituent words. For example, '#SpilledBeerOnFloor' is converted to 'Spilled Beer On Floor'. This is achieved using the *WordSegment* [1] module for word segmentation available in python. It is to be noted here that the segmented words are required only for obtaining word embeddings. For obtaining lexicon based features (cf. Section 2.3.1) the entire token with the '#' is used.

- **Elongation:** User tends to express their state of emotion by elongating a valid word e.g. '*jooooy*', '*goooodd*' etc. In this step, all such elongated words are identified and converted into valid words by removing the consecutive characters. For example '*jooyyyy*' and '*jooooy*' are converted to '*joy*'.

- **Verb present participle:** In Twitter domain, it is observed that user tends to omit the character '*i*' or '*g*' in words ending with 'ing'. For example, '*going*' is written as '*goin*' or '*gong*'. Such errors have been identified and corrected. We apply this rule for all the verbs that ends with either '*ng*' of '*in*'.

- **Frequent noisy term:** We compile a dictionary of frequently used slang terms and abbreviations along with its normal form that are commonly in practice in the Twitter domain. Every token in a tweet is searched in this dictionary. If a match is found then it is replaced with the normal form. The list was compiled utilizing the datasets of WNUT-2015 shared task on Twitter Lexical Normalization (Baldwin et al., 2015).

- **Expand contractions:** Contraction of a multi-word token is formed by making it shorter by dropping some characters and placing an apostrophe between them. For example, the contraction of 'i am' is 'iḿ'. We compile a dictionary of contractions and its normalized forms employing the datasets of (Baldwin et al., 2015). We replace every occurrence of a contraction in a tweet by its expanded form.

2.2 LSTM based Approach

Long short term memory (Hochreiter and Schmidhuber, 1997) network is a special kind of recurrent network that can efficiently learn sequences over a longer period of time. The proposed method utilizes LSTM network to obtain the sentence embedding vector, which is then fed as an input to SVR for prediction. The proposed network comprises of one Bidirectional LSTM (BiLSTM) (Schuster and Paliwal, 1997) layer followed by two dense layers. Hidden layer of the LSTMs consists of 100 neurons whereas the dense layers contain 100 and 50 neurons, respectively.

2.2.1 Word Embeddings

Word embedding (or word vector) is a distributed representation of words that contains syntactic and semantic information (Mikolov et al., 2013; Pennington et al., 2014). For this task, we use GloVe (Pennington et al., 2014) pre-trained word embedding trained on *common crawl* corpus. Each token in the tweet is represented by 300 dimension word vector. The choice of common crawl word embeddings for Twitter datasets is because of the normalization steps (Section 2.1). We observe that the application of normalization has a positive effect on the overall performance of the system.

[1] https://github.com/grantjenks/wordsegment

2.3 Particle Swarm Optimization based Feature Selection

Particle swarm optimization (Kennedy and Eberhart, 2001) is an optimization technique build over the social behavior of a flock of birds. Each potential solution, also known as particles, stores its best position attained so far. The global best solution recorded by any particle in the flock is also recorded and shared among the particles. In the search space, each particle moves towards the optimal solution based on its own best position and the global best position. Eventually, particles concentrate on a limited search space dictated by the global best solution found so far. The entire process is governed by three operations namely, *evaluate*, *compare* and *imitate*. Evaluation step quantifies the goodness of each particle, whereas, the comparison step obtains the best solution by comparing the particles. The imitate step produces new particles based on the best solution. A particle is an n-dimensional binary vector, where each element represents one feature. The value of each element (i.e. 0 or 1) signifies the presence or absence of its corresponding feature. Consequently, missing feature in a particle does not participate in training and testing of the system. On termination, PSO yields a particle (encoding a particular feature subset) that represents the best solution. We closely follow PSO based feature selection algorithm of (Akhtar et al., 2017) in the current work.

2.3.1 Feature Set

This section describes the features that we extract to predict the emotion intensity. All these features are fed to the PSO to generate the optimized feature set.

- **VADER Sentiment:** VADER (Gilbert, 2014) stands for Valence Aware Dictionary and Sentiment Reasoner. It is a rule-based sentiment analysis technique designed to work with contents on social media. For every input tweet, it provides positive, negative, neutral and compound sentiment score. We use these four values as features.

- **Lexicon based Features:** For each tweet we extract the following lexicon based features:
  - **Polar word count:** Count of positive and negative words using the *MPQA subjectivity lexicon* (Wiebe and Mihalcea, 2006) and *Bing Liu lexicon* (Ding et al., 2008).
  - **Aggregate polarity scores:** Positive and negative scores are obtained from each of the following lexicons: *Sentiment140* (Mohammad et al., 2013), *AFINN* (Nielsen, 2011) and *Sentiwordnet* (Baccianella et al., 2010). It is calculated by aggregating the positive and negative word scores provided by each lexicon.
  - **Aggregate polarity scores (Hashtags):** Aggregate of positive and negative scores of the hashtags in a tweet is calculated from *NRC Hashtag Sentiment lexicon* (Mohammad et al., 2013).
  - **Emotion word count:** Count of the number of words matching each emotion from *NRC Word-Emotion Association Lexicon* (Mohammad and Turney, 2013).
  - **Aggregate emotion score:** Sum of emotion associations of the words present in *NRC-10 Expanded lexicon* (Bravo-Marquez et al., 2016).
  - **Aggregate emotion score (Hashtags):** Sum of emotion associations of the hashtags in tweet matching the *NRC Hashtag Emotion Association Lexicon* (Mohammad and Kiritchenko, 2015).
  - **Emoticons score:** Positive and negative score of the emoticons obtained from *AFINN* project (Nielsen, 2011).
  - **Negation count:** Count of the number of negating words in the tweet.

2.4 Regression Model

An overall schema of the proposed system is depicted in Figure 1. Our proposed regression model consists of LSTM network and Support Vector Regression (SVR). First a LSTM network is trained using word vectors as input with *sigmoid* activation. Upon completion of training, the output of the top most hidden layer is used as *sentence embedding*. The trained sentence embeddings represent the relevant semantic and syntactic features of the tweets. Next, optimized feature set, as obtained by PSO, is concatenated with sentence embeddings for training a SVR model. The idea of cascading SVR with LSTM was motivated by the recent works of (Akhtar et al., 2016; Wang et al., 2016).

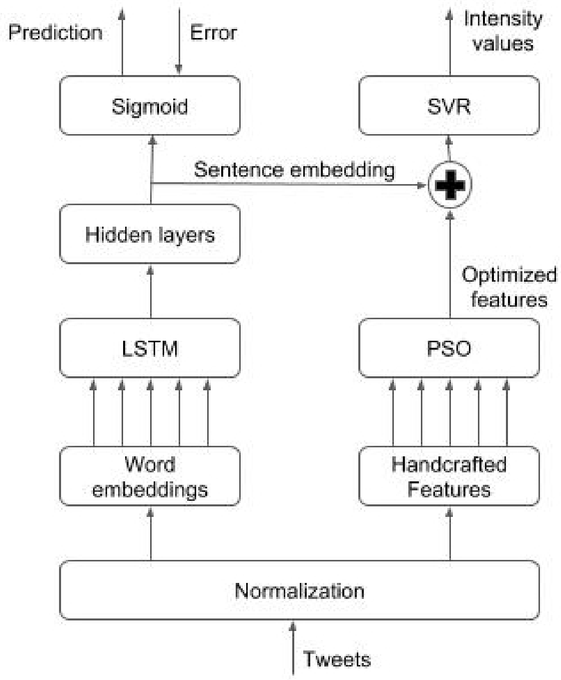

Figure 1: Proposed architecture.

3 Experiments, Results and Analysis

3.1 Dataset

The evaluation dataset (Mohammad and Bravo-Marquez, 2017) comprises of four emotions i.e. *anger*, *fear*, *sadness* and *joy*. The training set contains 857, 1147, 786 & 823 tweets for *anger, fear, sadness* and *joy*, respectively. The development set contains 84, 110, 74 & 79 tweets, while test set comprises of 760, 995, 673 & 714 tweets, respectively for each domain.

3.2 Experimental Results

We use Python based neural network library, i.e. Keras[2], for the implementation. For tokenization of tweets, we utilize CMU ARK tool[3]. The official evaluation metric was Pearson coefficient. We use *tanh* as an activation function at the intermediate layers while at the output layer we utilize *sigmoid*. We employ *Adam* (Kingma and Ba, 2014) optimizer and set the Dropout (Srivastava et al., 2014) as 40%. We train our network for 50 epochs. Table 1 depicts the evaluation results on the development and test sets. We first train a BiLSTM network utilizing GloVe common crawl embeddings. The resultant network produces average Pearson score of merely 0.1877. We observe that a good percentage of tokens (mostly noisy) were missing in the embeddings - thus poses challenge to the network during the learning phase. Subsequently, we try to minimize the effect of noisy tokens by utilizing GloVe Twitter embeddings. Though, the network obtains improved average Pearson score at 0.1921, improvement is not significant. On analysis we find similar issues with Twitter embeddings. To address the problem of data sparsity we employ a series of heuristics (c.f. Section 2.1) in order to normalize the text. Consequently, we obtain average Pearson score of 0.6289 with normalization outperforming the baseline system (0.610) provided by the organizers of the shared task.

We then cascade the LSTM network with SVR for the final predictions (*LSTM+SVR*). On cascading we obtain 0.6641 average Pearson score, reporting a gain of 0.04 points. Finally, to further improve the prediction accuracies we introduce various handcrafted lexicon features (c.f. Section 2.3.1) into the architecture (*LSTM+SVR+Feat*). Although, we see an improvement of 0.01 point in average Pearson score, introduction of same set of lexicons features have contrasting effect on different emotion datasets i.e. *anger*, *fear*, *joy* & *sadness*. We observe improvement for *joy* and *sadness*, whereas for *anger* use of this same set of features degrades the system performance. For *fear*, introduction of features to *LSTM+SVR* almost have no effect. Motivated by these results we perform PSO based feature selection algorithm in order to find optimal set of features for different emotions. We get the best average Pearson score of 0.7271 on the development set by utilizing sentence embeddings, optimized feature set and SVR (*LSTM+SVR+PSO*). We also observe improvement in Pearson score for each of the emotion datasets ranging from 0.5-0.7 points over *LSTM+SVR*. It is evident from the obtained results that normalization of tweets is a major factor in obtaining good performance. Also, introduction of the PSO based feature selection in *LSTM+SVR* hybrid model further assists the system in improving the performance.

On final evaluation, i.e. on the test set, our proposed system (*LSTM+SVR+PSO*) scores an average Pearson score of 0.682. In comparison, baseline system produces 0.6470 average Pearson

[2] http://keras.io/

[3] http://www.cs.cmu.edu/~ark/

Models	Descriptions	Pearson score				
		Anger	Fear	Joy	Sadness	Avg
RESULT ON DEV SET						
Sentence embeddings - Normalization*	*LSTM*	0.178	0.029	0.462	0.080	0.187
Sentence embeddings - Normalization*$^{\#}$	*LSTM*	0.153	0.050	0.462	0.101	0.192
Sentence embeddings	*LSTM*	0.629	0.645	0.737	0.504	0.628
Sentence embeddings	*LSTM+SVR*	0.669	0.661	0.761	0.563	0.664
Sentence embeddings + All features	*LSTM+SVR+Feat*	0.610	0.663	0.806	0.611	0.673
Sentence embeddings + PSO	***LSTM+SVR+PSO***	**0.719**	**0.732**	**0.826**	**0.632**	**0.727**
Baseline (Mohammad and Bravo-Marquez, 2017)	*LibLinear*	0.599	0.580	0.694	0.569	0.610
RESULT ON TEST SET						
Sentence embeddings + PSO	***LSTM+SVR+PSO***	**0.649**	**0.713**	**0.657**	**0.709**	**0.682**
Baseline (Mohammad and Bravo-Marquez, 2017)	*LibLinear*	0.625	0.620	0.635	0.706	0.647

Table 1: Evaluation results on development and test set. *Without normalization step; Other models are with normalization. $^{\#}$With GloVe Twitter word embeddings; Other models utilize GloVe common crawl embeddings.

Lexicons	Datasets			
	Anger	Fear	Joy	Sadness
MPQA			✓	✓
Bing Liu		✓		
SentiWordNET		✓	✓	✓
AFINN			✓	
Sentiment140			✓	✓
NRC Hashtag Sentiment		✓	✓	✓
NRC Hashtag Emotion	anger	anger, anticipation, fear & surprise	anticipation, joy, sadness & surprise	disgust & sadness
NRC10 Expanded	anger, disgust, surprise, positive, anger-ex, fear-ex, positive-ex, negative-ex	anticipation, joy, sadness, surprise, positive, negative, fear-ex, disgust-ex, surprise-ex	anticipation, joy, trust, joy-ex, surprise-ex	anger, anticipation, disgust, fear, surprise, anticipation-ex, disgust-ex, fear-ex, surprise-ex, negative-ex
Emoticons-AFINN		✓		✓

Table 2: Optimized feature set for four datasets.

score, a difference of 4%. For *anger* and *fear* we observe a small performance drop on the test set as compared to the development set while our proposed system performs better in case of *sadness*. Further, we observe that our system does not perform at par (a drop of nearly 17%) for *joy* as compared to the development set. However, similar phenomenon was observed for the baseline system as well i.e. a drop of 6% in *joy*. We also observe that our proposed system is statistically significant over baseline system with p-value = 0.03683.

Table 2 shows the optimized set of feature for four datasets i.e. *anger*, *fear*, *joy* and *sadness*. It is evident from the table that some of the features have high degree of relevance than others. For example, NRC Hashtag Emotion (Mohammad and Kiritchenko, 2015) & NRC10 Expanded (Bravo-Marquez et al., 2016) lexicons have been utilized by all four of them, whereas Bing Liu (Ding et al., 2008) and AFINN (Nielsen, 2011) lexicons have been employed by only *fear* & *joy*, respectively.

3.3 Error Analysis

We also perform error analysis on the obtained results. Following are the few cases where our system consistently suffers in predicting the intensity values.

- Presence of high intensity emotion words (such as anger, revenge, fury, exciting etc) makes it non-trivial for the system to correctly predicts the intensity values.

  Example 1:
  Tweet: #Forgiveness might make us look #weak, but the weakest person is the one who holds #anger, #hatred, and #revenge.
  Actual: 0.354 **Predicted:** 0.630

  Example 2:
  Tweet: Police: Atlanta rapper Shawty Lo killed in fiery car crash.
  Actual: 0.396 **Predicted:** 0.619

4 Conclusion

In this paper, we have presented a hybrid LSTM-SVR architecture for predicting the intensity level *w.r.t.* to an emotion. We first applied various heuristics for normalizing the tweets. Following this step, the noisiness of tweets is addressed to a great effect and consequently improves the performance of the system. The proposed approach further utilized relevant set of hand-crafted features obtained through a PSO based feature selection technique. Adding optimized features in the proposed architecture (*LSTM+SVR+PSO*) attains significant improvement over the system without it (*LSTM+SVR*) and this phenomenon was observed for all the four emotion datasets i.e. *anger*, *fear*, *joy* and *sadness*.

References

Md Shad Akhtar, Deepak Gupta, Asif Ekbal, and Pushpak Bhattacharyya. 2017. Feature selection and ensemble construction: A two-step method for aspect based sentiment analysis. *Knowledge-Based Systems* 125:116 – 135.

Md Shad Akhtar, Ayush Kumar, Asif Ekbal, and Pushpak Bhattacharyya. 2016. A hybrid deep learning architecture for sentiment analysis. In *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan*. pages 482–493.

Md Shad Akhtar, Utpal Kumar Sikdar, and Asif Ekbal. 2015. IITP: Hybrid Approach for Text Normalization in Twitter. In *Proceedings of the ACL 2015 Workshop on Noisy User-generated Text (WNUT-2015*. Beijing, China, pages 106–110.

Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. 2010. SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining. In *LREC*. volume 10, pages 2200–2204.

Timothy Baldwin, Marie Catherine de Marneffe, Bo Han, Young-Bum Kim, Alan Ritter, and Wei Xu. 2015. Shared Tasks of the 2015 Workshop on Noisy User-generated Text: Twitter Lexical Normalization and Named Entity Recognition. In *Proceedings of the ACL 2015 Workshop on Noisy User-generated Text*. Beijing, China, pages 126–135.

Felipe Bravo-Marquez, Eibe Frank, Saif M Mohammad, and Bernhard Pfahringer. 2016. Determining word–emotion associations from tweets by multi-label classification. In *WI'16*. IEEE Computer Society, pages 536–539.

Xiaowen Ding, Bing Liu, and Philip S Yu. 2008. A Holistic Lexicon-based Approach to Opinion Mining. In *Proceedings of the 2008 international conference on web search and data mining*. ACM, pages 231–240.

Paul Ekman. 1992. An argument for basic emotions. *Cognition and Emotion* pages 169–200.

CJ Hutto Eric Gilbert. 2014. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Eighth International Conference on Weblogs and Social Media (ICWSM-14). Available at (20/04/16) http://comp. social. gatech. edu/papers/icwsm14. vader. hutto. pdf*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9(8):1735–1780.

James Kennedy and Russell C. Eberhart. 1995. Particle swarm optimization. In *Proceedings of the IEEE International Conference on Neural Networks*. pages 1942–1948.

James Kennedy and Russell C. Eberhart. 2001. *Swarm Intelligence*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR* abs/1412.6980. http://dblp.uni-trier.de/db/journals/corr/corr1412.html.

Shih-Wei Lin, Kuo-Ching Ying, Shih-Chieh Chen, and Zne-Jung Lee. 2008. Particle swarm optimization for parameter determination and feature selection of support vector machines. *Expert systems with applications* 35(4):1817–1824.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. *arXiv preprint arXiv:1301.3781* .

Saif Mohammad, Svetlana Kiritchenko, and Xiaodan Zhu. 2013. NRC-Canada: Building the State-of-the-Art in Sentiment Analysis of Tweets. In *Proceedings of the seventh international workshop on Semantic Evaluation Exercises (SemEval-2013)*. Atlanta, Georgia, USA.

Saif M. Mohammad and Felipe Bravo-Marquez. 2017. WASSA-2017 Shared Task on Emotion Intensity. In *Proceedings of the Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis (WASSA)*. Copenhagen, Denmark.

Saif M Mohammad and Svetlana Kiritchenko. 2015. Using hashtags to capture fine emotion categories from tweets. *Computational Intelligence* 31(2):301–326.

Saif M. Mohammad and Peter D. Turney. 2013. Crowdsourcing a Word-Emotion Association Lexicon 29(3):436–465.

Finn Årup Nielsen. 2011. A new ANEW: Evaluation of a word list for sentiment analysis in microblogs. *arXiv preprint arXiv:1103.2903* .

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*. pages 1532–1543. http://www.aclweb.org/anthology/D14-1162.

Rosalind W. Picard. 1997. *Affective Computing*. MIT Press, Cambridge, MA, USA.

M. Schuster and K.K. Paliwal. 1997. Bidirectional Recurrent Neural Networks. *Trans. Sig. Proc.* 45(11):2673–2681.

Alex J. Smola and Bernhard Schölkopf. 2004. A tutorial on support vector regression. *Statistics and Computing* 14(3):199–222. https://doi.org/10.1023/B:STCO.0000035301.49549.88.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* 15:1929–1958. http://jmlr.org/papers/v15/srivastava14a.html.

Wenya Wang, Sinno Jialin Pan, Daniel Dahlmeier, and Xiaokui Xiao. 2016. Recursive Neural Conditional Random Fields for Aspect-based Sentiment Analysis. *CoRR* abs/1603.06679. http://arxiv.org/abs/1603.06679.

Janyce Wiebe and Rada Mihalcea. 2006. Word Sense and Subjectivity. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, pages 1065–1072.

Shweta Yadav, Asif Ekbal, Sriparna Saha, and Pushpak Bhattacharyya. 2017. Entity Extraction in Biomedical Corpora: An Approach to Evaluate Word Embedding Features with PSO based Feature Selection. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*. Valencia, Spain, page 11591170.