AffecThor at SemEval-2018 Task 1: A cross-linguistic approach to sentiment intensity quantification in tweets

Mostafa Abdou and **Artur Kulmizev** and **Joan Ginés i Ametllé**
{m.abdou, a.kulmizev, j.gines.i.ametlle}@student.rug.nl
CLCG, University of Groningen

Abstract

In this paper we describe our submission to SemEval-2018 Task 1: *Affects in Tweets*. The model which we present is an ensemble of various neural architectures and gradient boosted trees, and employs three different types of vectorial tweet representations. Furthermore, our system is language-independent and ranked first in 5 out of the 12 subtasks in which we participated, while achieving competitive results in the remaining ones. Comparatively remarkable performance is observed on both the Arabic and Spanish languages.

1 Introduction

The *Affects in Tweets* shared task (Mohammad et al., 2018) is the second iteration of a task which offers a new approach to Sentiment Analysis - one that concerns itself with emotion and sentiment *intensity*, rather than simple categorical classification. The shared task is divided into a set of subtasks, where the aim is to predict the emotion intensity of a predetermined emotion (fear, anger, sadness, joy) or sentiment (valence) intensity of a given set of tweets. Such predictions are either formulated as a regression problem where the output is a continuous-valued score in the interval $(0, 1)$, or as ordinal classification into a given number of classes representing intensity. Additionally, each one of the subtasks targets a particular language: English, Arabic or Spanish.

In total, we participated in 12 different subtasks and our system achieved the best performance on the test set out of all participants in 5 out of those, ranked second in 3 others, and performed competitively in the rest. Moreover, our system can arguably be considered the best overall performing system for both Arabic and Spanish[1]. It should be noted, however, that the shared task includes traditional emotion classification subtasks in which we did not participate.

The system described in this paper builds upon a survey of some of the best performing systems from previous related shared tasks (Mohammad and Bravo-Marquez, 2017; Rosenthal et al., 2017). In particular, we draw inspiration from the systems described in (John and Vechtomova, 2017), which makes use of gradient boosted trees for regression; (Goel et al., 2017), which employs an ensemble of various neural models; and (Baziotis et al., 2017), which features Long Short Term Memory (LSTM) networks with an attention mechanism. Our work contributes to the aforementioned approaches by further developing a variety of neural architectures, using transfer learning via pretrained sentence encoders, testing methods of ensembling neural and non-neural models, and gauging the performance and stability of a regressor across languages.

The rest of this paper describes the pipeline of the system used for our submission, which is an ensemble of neural and non-neural models.

2 Data and features

The provided training and development data is comprised of tweets, an emotion or sentiment, and labels describing the intensity of the emotion or

[1] https://competitions.codalab.org/competitions/17751#results

Proceedings of the 12th International Workshop on Semantic Evaluation (SemEval-2018), pages 210–217
New Orleans, Louisiana, June 5–6, 2018.

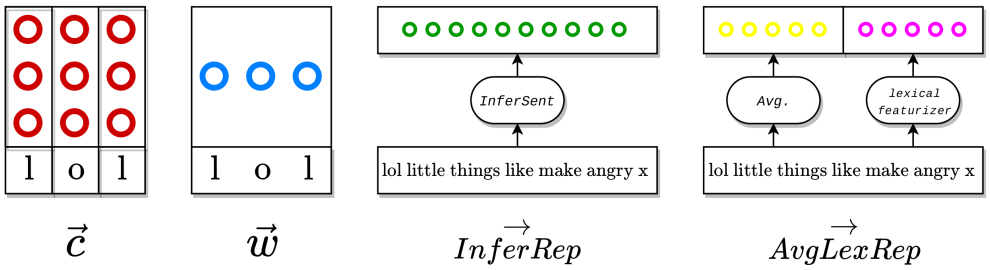

Figure 1: Graphical visualization of various feature vectors used in our ensemble model. These are from left to right: character embedding, word embedding, $\overrightarrow{InferRep}$ and $\overrightarrow{AvgLexRep}$ representations.

sentiment. We refer readers interested in an exhaustive description of the data to (Mohammad et al., 2018; Mohammad and Kiritchenko, 2018). In this work, we convert each tweet into a combination of three types of vector representations: character and word-level vectors for Arabic and Spanish; and character, word, and sentence-level vectors for English. This section describes the procedure that allows us to obtain these varied representations, which are later employed by our classification and regression models.

2.1 Preprocessing

The syntactic and orthographic form of tweets often differs substantially from text belonging to other domains (John and Vechtomova, 2017). As such, pre-processing procedures are as important as the architecture of any given model.

In pre-processing our data, we first replace all same-character sequences of length 3 or more with only 2 occurrences. We also replace all user mentions with a unique common token, as well as all control characters with whitespaces. Emojis are surrounded with spaces, enforcing that any two emojis are not consecutive characters. Finally, all text is lowercased. In the case of Spanish text, we further remove the characters ¿ and ¡, and replace accented characters with their unaccented versions, as well as ñ with *n*. In the case of Arabic text, we remove quotation marks as well.

Following the cleaning process, we tokenize the resulting text by applying the `twokenize` tool (Krieger and Ahn, 2010), as provided in the CMU Tweet NLP software (Owoputi et al., 2013), which is, by design, able to cope with the noise that appears in social media. Once the tokenization is completed, we filter all stopwords [2].

[2]We employ the stopword lists available from `https://www.ranks.nl/stopwords`

2.2 Lexicons

Lexicons are one of the resources which we employ in order to compute features. In short, a lexicon is a collection of words that are associated with a value for an arbitrary number of affective categories. In our case, given a tweet, we produce several features per lexicon which are the result of aggregating individual matching word values in each category, adding the numerical values and counting those which are nominal. We provide an overview of the lexicons used per language below, with the number of features contributed by each individual lexicon in parenthesis. In the case of English, the following lexicons and extracted values jointly produce a feature vector of dimension 43:

- MPQA lexicon (2): Number of positive and negative words (Wilson et al., 2005).
- Bing Liu lexicon (2): Number of positive and negative words (Hu and Liu, 2004).
- Emoticons (2): Positive and negative aggregated scores for emoticons (Nielsen, 2011).
- Sentiment140 lexicon (2): Positive and negative aggregated scores (Kiritchenko et al., 2014).
- NRC Word-Emotion Association Lexicon (10): Number of words matching each category (Mohammad and Turney, 2013).
- NRC Hashtag Sentiment lexicon (2): Positive and negative aggregated scores (Kiritchenko et al., 2014).
- NRC Hashtag Emotion Association Lexicon (8): Aggregated scores for each category (Mohammad and Kiritchenko, 2015).

- NRC-10-Expanded lexicon (10): Aggregated scores for each category (Bravo-Marquez et al., 2016).
- SentiWordnet (2): Positive and negative aggregated scores (Baccianella et al., 2010).
- AFINN lexicon (2): Positive and negative aggregated scores (Nielsen, 2011).
- Negations (1): Number of negative words (Mohammad and Bravo-Marquez, 2017).

In the case of Arabic we also employ the same first 6 lexicons which we listed for English, but with the content words automatically translated (Salameh et al., 2015). However, we extract 4 scores from the MPQA lexicon (on the affective categories *positive*, *negative*, *neutral* and *both*), an a single combined score from the Bing Liu and Emoticons lexicons. Furthermore, we employ 3 lexicons generated by distant supervision techniques on Arabic tweets as follows (Mohammad et al., 2016), in order to obtain a feature vector of dimension 26:

- Arabic Emoticon Lexicon (2): Number of positive and negative words.
- Arabic Hashtag Lexicon (2): Number of positive and negative words.
- Arabic dialectal Hashtag Lexicon (2): Number of positive and negative words.

Finally, the following lexicons are used in Spanish to produce a feature vector of dimension 14. In contrast to the Arabic language, the majority of the lexicons here listed are manually annotated or semi-automatically generated from Spanish data:

- Emoticons (1): Combination of positive and negative aggregated scores for emoticons (Nielsen, 2011).
- El Huyar dictionary (2): Positive and negative aggregated scores (Saralegi and San Vicente, 2013).
- ISOL lexicon (2): Number of positive and negative words (Martínez-Cámara et al., 2014).
- SDAL lexicon (3): Aggregated scores for each category (Dell' Amerlina Ríos and Gravano, 2013).
- Spanish Sentiment lexicon (2): Number of positive and negative words (Perez Rosas et al., 2012).
- ML Senticon (1): Aggregated score for polarity (Cruz et al., 2014).
- Sentwords (3): Aggregated score for each category in an automatically translated version of the lexicon described in (Beth Warriner et al., 2013).

Note that the lexicons are not directly used on tweet data, but rather that lexical features are extracted after applying the same data cleaning and tokenization process which we described for the training data to each one of the lexicons listed.

2.3 Word embeddings

Word embeddings are another popular choice for feature extraction. We employ pre-trained word embeddings for English and train our own embeddings on separated Arabic and Spanish tweet data that we manually collected. All sets of embeddings comprise 400 dimensions and are detailed below for each language:

- English: Word2vec skip-gram embeddings, trained on the Edinburgh Twitter Corpus (Petrović et al., 2010).
- Arabic: Word2vec skip-gram embeddings, trained on 4.38 million tweets[3].
- Spanish: Word2vec skip-gram embeddings, trained on 3.02 million tweets[4].

2.4 Manually-crafted representations

In the Arabic and Spanish subtasks, some model components in our ensemble use a combination of the two types of representations described so far (lexical features and word embeddings) as an input feature vector. To obtain this, we average the embeddings corresponding to each word in a given tweet up to a maximum of 25 words, and append the computed lexical features to the result. These features are extracted using the filters provided in the Affective Tweets package (Mohammad and Bravo-Marquez, 2017) available for WEKA (Hall et al., 2009). In this paper, we will refer to this combined representation as $Avg\vec{Lex}Rep$.

[3]Available for download from akulmizev.com/embeddings/ar_tweets.csv.

[4]Available for download from akulmizev.com/embeddings/es_tweets.csv.

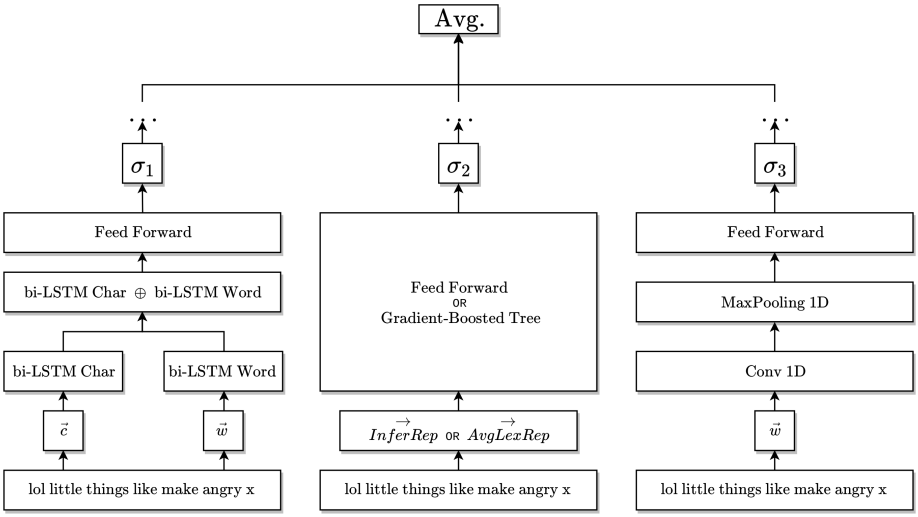

Figure 2: Diagram of our system which describes how the different models used are ensembled. The outputs of each component in the ensemble are averaged into a single score.

2.5 Learned representations

Engineering a representation of the data (such as a the one described in Section 2.4) that can support effective machine learning is a complex task, requiring human ingenuity and domain-specific knowledge. Representation learning techniques (Bengio et al., 2003) enable machine learning algorithms to automatically extract and organize discriminative features, thereby mapping raw data into forms that make it easier to extract useful information. Some model components in our ensemble employ this kind of representation, which we obtain using 2 different methods:

- Encoding a tweet using (Conneau et al., 2017)'s BiLSTM-max pooling encoder, which is pre-trained on a natural language inference dataset[5] and produces representations that perform well on a wide variety of NLP tasks. This approach in particular employs GloVe word embeddings (Pennington et al., 2014) as input and produces a vector containing 4096 dimensions, to which we will refer with the name $\vec{InferRep}$. However, note that we only produce this feature vector for the English language subtasks.

- Encoding a tweet using one or a combination of three neural architectures which use skip-gram word embeddings (Mikolov et al., 2013) as input and are trained on the shared task's training data for regression subtasks. These correspond to the CNN, Bi-LSTM and CHAR-LSTM models described in Section 3. Representations produced by the CHAR-LSTM model are of dimension 612[6], and the ones obtained via the Bi-LSTM model are of dimension 512. Representations produced by the CNN model have different dimensionality depending on the number and size of filters used. We will collectively refer to such representations with the name $\vec{RegRep}$.

3 System architecture

While $\vec{RegRep}$ is produced as part of end-to-end trainable regression and classification models, $\vec{AvgLexRep}$ and $\vec{InferRep}$ are generated independently. Thus, $\vec{AvgLexRep}$ and $\vec{InferRep}$ are fed separately into these models after being generated. The pipeline of our ensemble is represented schematically in Figure 2.

3.1 Neural models

We implement three varieties of neural network architectures which are commonly used in text classification tasks using Keras (Chollet et al., 2015) with a TensorFlow backend. In all of them, our objective function is Mean Squared Error (MSE) and dropout (Srivastava et al., 2014) is used for regularization at various levels. These architectures are listed below:

[5] Stanford Natural Language Inference dataset (Bowman et al., 2015). This is only available for English.

[6] 512 dimensions correspond to word final hidden states and 100 dimensions to character hidden states.

- Convolutional Neural Network (CNN) with max pooling.
- Bidirectional Long-Short Term Memory (Bi-LSTM) with attention.
- Combined character and word features bi-LSTMs (CHAR-LSTM).

3.2 Regression

For *AvgLexRep* and *InferRep*, which are not part of an end-to-end trainable model, we perform regression using either a feed-forward Deep Neural Network (DNN) or Gradient Boosted Trees (GBT)[7]. The depth of the feed-forward network is determined constructively, starting with one layer and adding layers which are half the size of the previous one until performance on cross-validation stops improving.

3.3 Model selection for regression

We perform model selection using 5-fold cross-validation on the training data from the shared task. In each subtask that involves regression, the possible models are ranked according to their individual performance and ensembled through simple averaging. The ensemble itself is built constructively based on the ordering defined by the ranking, starting from a single component and adding components in order whenever the average performance on cross-validation improves.

Ensembling has long been shown to be an effective method of variance reduction for complex models (Perrone, 1993), and we indeed find in our experiments that averaging predictions leads to results better than those of any individual model[8].

Furthermore, we also find predictions obtained via simple averaging to be more accurate (on cross-validation) compared to those obtained via feeding the outputs from all model components into a sigmoid layer. Although such a finding might appear counter-intuitive, it can perhaps be explained through the fact that the training dataset is relatively small, and therefore ensembling via a non-linear function of the outputs can potentially lead to overfitting.

[7]We use the GBT implementation provided in scikit-learn (Pedregosa et al., 2011).

[8]We refer the reader to (Hashem and Schmeiser, 1993) for an explanation of why this is the case.

3.4 Ordinal classification

Our system for each ordinal classification subtask makes use of the ensemble model which we build for the corresponding regression subtask in the same language, and model selection is performed using the same procedure described in Section 3.3. However, instead of averaging the predictions, the best model's predictions are concatenated and fed as features to an ordinal meta-classifier (Antoniuk et al., 2013).

3.5 Hyperparameter tuning

Hyper-parameter optimization is carried out using 5-fold cross-validation. At first, a reasonable range is determined manually, and then grid-search is performed within that range. For Gradient Boosted Trees, the hyper-parameters optimized are maximum tree depth, number of estimators, and maximum leaf nodes. For neural models, the parameters optimized are batch size, number of epochs, size of the layers or filters, and whether or not dropout is used at different levels. Dropout is by default always set at 0.2. Furthermore, we use a fixed random seed to enable replicability.

4 Evaluation

System	Anger		Sadness		Joy		Fear	
	CV	Test	CV	Test	CV	Test	CV	Test
DNN (Infer.)	0.707	0.703	0.755	0.654	0.713	0.667	0.742	0.701
GBT (Infer.)	0.716	0.707	0.739	0.677	0.708	0.688	0.748	0.697
CHAR-LSTM	0.698	0.682	0.716	0.626	0.722	0.700	0.727	0.663
CNN	0.642	0.636	0.521	0.4316	0.637	0.628	0.615	0.459
Ensemble	0.756	0.749	0.770	0.699	0.758	0.740	0.773	0.726

Table 1: Comparison of Pearson correlation cross-validation (CV) and official results (Test) scores in the Emotion Intensity regression (EI-reg) English subtasks. Results are given for both the ensemble and its individual models.

Table 1 displays the scores (both 5-fold cross-validation and test scores) of the individual models and the ensemble model for the Emotion Intensity English regression subtasks. The ensemble model in this case is always for the best three models. Table 2 shows the results obtained using 5-fold cross-validation on the combined training and development data and the official test set results for each subtask. All scores are reported as the Pearson correlation coefficient between our system's predictions and the provided gold-labels (i.e. human judgments).

Task	Emotion	English		Arabic		Spanish	
		CV	Test	CV	Test	CV	Test
El-reg	Anger	0.756	0.749	0.620	0.647	0.731	0.676
	Joy	0.758	0.740	0.690	0.756	0.712	0.753
	Fear	0.773	0.726	0.619	0.642	0.720	0.776
	Sadness	0.770	0.669	0.717	0.694	0.728	0.746
	Macro-avg.	0.764	0.728	0.662	0.685	0.723	0.738
V-reg	Valence	0.800	0.829	0.820	0.816	0.775	0.795
El-oc	Anger	0.670	0.620	0.620	0.551	0.635	0.606
	Joy	0.701	0.686	0.610	0.631	0.668	0.667
	Fear	0.635	0.528	0.565	0.551	0.658	0.706
	Sadness	0.738	0.622	0.682	0.618	0.655	0.677
	Macro-avg.	0.691	0.616	0.619	0.587	0.654	0.664
V-oc	Valence	0.770	0.776	0.778	0.752	0.749	0.756

Table 2: Pearson correlation using cross-validation (CV) on the trainining data and official results of the shared task (Test) obtained with our system, for each one of the Emotion Intensity (EI), Valence (V), regression (reg) and ordinal classification (oc) subtasks.

5 Analysis

It can be observed in Table 2 that the test and cross-validation scores are similar, meaning that cross-validation provided an accurate estimate of the generalization error and that our system's overfitting of the different combined training and development sets is minimal. In fact, for the English valence subtasks, the Arabic Emotion Intensity regression subtask and all Spanish subtasks except the ones involving anger as the target emotion, the test scores are higher or equal than the cross-validation scores. This indicates both that our system generalizes appropriately and that the test sets are not substantially different than the training sets.

Overall performance is higher for English, likely due to the availability of better quality lexicons and word embeddings. Nonetheless, it is interesting to note that on average, cross-validation provided an optimistic estimate of the generalization error for English and a pessimistic one for Spanish and Arabic.

Furthermore, as shown in Table 1 for various English regression subtasks, it is clear that the ensemble outperforms all individual models on both cross-validation and the test set. This points towards the success of our ensembling method in reducing the variance of individual models. We omit similar results for other subtasks because the trend displayed by those is comparable.

Finally, it is interesting to note that the models using $\vec{InferRep}$ (DNN and GBT), which rely on tweet representations produced through transfer learning from Natural Language Inference, outperformed the models using the task-specific $\vec{RegRep}$ (CNN, Bi-LSTM and CHAR-LSTM) for all emotions except Sadness.

6 Conclusion and future work

In this paper we have described *AffecThor*, the system which we submitted to the SemEval-2018 *Affects in Tweets* shared task. *AffecThor* uses three different types of learned and manually-crafted representations and is an ensemble of neural and non-neural models. It is the best performing system on 5 out of 12 subtasks, and the second best performing in 3 others. Furthermore, it is arguably the best overall performer for Spanish and Arabic.

Our work explored two methods of ensembling regressors: simple averaging and using a non-linearity (sigmoid) layer on top of the different sub-models as part of an end-to-end trainable neural model, and found that simple averaging is more robust. However, we believe that ensembling using a linear combination (weighted-averaging) where the weights are learned could lead to better results, as is shown in (Perrone, 1993; Hashem and Schmeiser, 1993).

Finally, the availability of fine-grained labeled data across emotions and languages opens up the possibility of investigating multi-task and multilingual learning objectives. In the future, we would like to extend this work in that direction.

References

Kostiantyn Antoniuk, Vojtěch Franc, and Václav Hlaváč. 2013. Mord: Multi-class Classifier for Ordinal Regression. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 96–111. Springer.

Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. 2010. Sentiwordnet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining. In *Proceedings of the International Conference on Language Resources and Evaluation*.

Christos Baziotis, Nikos Pelekis, and Christos Doulkeridis. 2017. Datastories at SemEval-2017 Task 4: Deep LSTM with Attention for Message-level and Topic-based Sentiment Analysis. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 747–754.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A Neural Probabilistic Language Model. *J. Mach. Learn. Res.*, 3:1137–1155.

Amy Beth Warriner, Victor Kuperman, and Marc Brysbaert. 2013. Norms of valence, arousal, and dominance for 13,915 English lemmas. 45:1191–1207.

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*.

Felipe Bravo-Marquez, Eibe Frank, Saif M. Mohammad, and Bernhard Pfahringer. 2016. Determining Word-Emotion Associations from Tweets by Multi-label Classification. In *2016 IEEE/WIC/ACM International Conference on Web Intelligence*, pages 536–539.

François Chollet et al. 2015. Keras. `https://github.com/keras-team/keras`.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loic Barrault, and Antoine Bordes. 2017. Supervised Learning of Universal Sentence Representations from Natural Language Inference Data. *arXiv preprint arXiv:1705.02364*.

Fe.L. Cruz, J.A. Troyano, Beatriz Pontes, and F. Javier Ortega. 2014. ML-SentiCon: A multilingual, lemma-level sentiment lexicon. 53:113–120.

Matías Dell' Amerlina Ríos and Agustín Gravano. 2013. Spanish DAL: A Spanish Dictionary of Affect in Language. In *Proceedings of the 4th Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis (WASSA 2013)*, pages 21–28.

Pranav Goel, Devang Kulshreshtha, Prayas Jain, and Kaushal Kumar Shukla. 2017. Prayas at EmoInt 2017: An Ensemble of Deep Neural Architectures for Emotion Intensity Prediction in Tweets. In *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 58–65.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. The WEKA Data Mining Software: An Update. *SIGKDD Explor. Newsl.*, 11(1):10–18.

Sherif Hashem and Bruce Schmeiser. 1993. Approximating a Function and its Derivatives Using MSE-Optimal Linear Combinations of Trained Feedforward Neural Networks. In *In Proceedings of the Joint Conference on Neural Networks*, pages 617–620.

Minqing Hu and Bing Liu. 2004. Mining and Summarizing Customer Reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 168–177.

Vineet John and Olga Vechtomova. 2017. UWatEmote at EmoInt-2017: Emotion Intensity Detection using Affect Clues, Sentiment Polarity and Word Embeddings. In *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 249–254.

Svetlana Kiritchenko, Xiaodan Zhu, and Saif M. Mohammad. 2014. Sentiment Analysis of Short Informal Texts. *Journal of Artificial Intelligence Research (JAIR)*, 50:723–762.

Michel Krieger and David Ahn. 2010. TweetMotif: Exploratory Search and Topic Summarization for Twitter. In *In Proceedings of AAAI Conference on Weblogs and Social Media*.

Eugenio Martínez-Cámara, M. Teresa Martín-Valdivia, M. Dolores Molina-González, and José M. Perea-Ortega. 2014. Integrating Spanish Lexical Resources by Meta-classifiers for Polarity Classification. *J. Inf. Sci.*, 40(4):538–554.

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 Felipe Bravo-Marquez. 2017. Emotion Intensities in Tweets. In *Proceedings of the 6th Joint Conference on Lexical and Computational Semantics, *SEM @ACM 2017, Vancouver, Canada, August 3-4, 2017*, pages 65–77.

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, Felipe Bravo-Marquez, Mohammad Salameh, and Svetlana Kiritchenko. 2018. Semeval-2018 Task 1: Affect in tweets. In *Proceedings of International Workshop on Semantic Evaluation (SemEval-2018)*, New Orleans, LA, USA.

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 Svetlana Kiritchenko. 2018. Understanding Emotions: A Dataset of Tweets to Study Interactions between Affect Categories. In *Proceedings of the 11th Edition of the Language Resources and Evaluation Conference*, Miyazaki, Japan.

Saif M. Mohammad, Mohammad Salameh, and Svetlana Kiritchenko. 2016. Sentiment Lexicons for Arabic Social Media. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*. European Language Resources Association (ELRA).

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

Årup Nielsen. 2011. A New ANEW: Evaluation of a Word List for Sentiment Analysis in Microblogs. In *Proceedings of the ESWC2011 Workshop on 'Making Sense of Microposts': Big things come in small packages*, pages 93–98.

Olutobi Owoputi, Chris Dyer, Kevin Gimpel, Nathan Schneider, and Noah A. Smith. 2013. Improved Part-Of-Speech Tagging for Online Conversational Text with Word Clusters. In *In Proceedings of NAACL*.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Veronica Perez Rosas, Carmen Banea, and Rada Mihalcea. 2012. Learning Sentiment Lexicons in Spanish. In *Proceedings of the international conference on Language Resources and Evaluation (LREC)*.

Michael Peter Perrone. 1993. *Improving Regression Estimation: Averaging Methods for Variance Reduction with Extensions to General Convex Measure Optimization*. Ph.D. thesis.

Saša Petrović, Miles Osborne, and Victor Lavrenko. 2010. The Edinburgh Twitter Corpus. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Linguistics in a World of Social Media*, pages 25–26.

Sara Rosenthal, Noura Farra, and Preslav Nakov. 2017. SemEval-2017 Task 4: Sentiment Analysis in Twitter. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 502–518.

Mohammad Salameh, Saif M. Mohammad, and Svetlana Kiritchenko. 2015. Sentiment after Translation: A Case-Study on Arabic Social Media Posts. In *HLT-NAACL*, pages 767–777. The Association for Computational Linguistics.

Xabier Saralegi and Iaki San Vicente. 2013. *Workshop on Sentiment Analysis at SEPLN (TASS2013)*.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis. In *HLT/EMNLP 2005, Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 347–354.