EiTAKA at SemEval-2018 Task 1: An Ensemble of N-Channels ConvNet and XGboost Regressors for Emotion Analysis of Tweets

Mohammed Jabreel **Antonio Moreno**
Intelligent Technologies for Advanced Knowledge Acquisition (ITAKA),
Departament d'Enginyeria Informàtica i Matemàtiques,
Universitat Rovira i Virgili,
Av. Països Catalans, 26, 43007 Tarragona, Spain
<first_name>.<last_name>@urv.cat

Abstract

This paper describes our system that has been used in Task1 Affect in Tweets. We combine two different approaches. The first one called N-Stream ConvNets, which is a deep learning approach where the second one is XGboost regressor based on a set of embedding and lexicons based features. Our system was evaluated on the testing sets of the tasks outperforming all other approaches for the Arabic version of valence intensity regression task and valence ordinal classification task.

1 Introduction

Sentiment Analysis is the task of automatically identifying the valence or polarity of a piece of text. This piece of text can be a user review, a document, an SMS message, a tweet, etc. According to (Mohammad, 2016), the term sentiment analysis also refers to determining the attitude towards a particular target or topic. The attitude can be the polarity (positive or negative), or an emotional or effectual attitude such as joy, anger, sadness and so on.

Most of the researchers in sentiment analysis have focused on developing systems to determine the polarity of a given text. This involves designing classifiers based on a set of examples with a manually annotated sentiment polarity. Although developing systems that automatically determine the intensity (i.e. the degree or the amount) of emotions that are communicated in a text has a wide range of applications in commerce, public health, social welfare, etc., most of the work has focused on categorical classification (whether a given piece of text communicates anger, joy, sadness, etc.). This can be attributed to the lack of suitable annotated data (Mohammad and Bravo-Márquez, 2017) .

In task1: *Affect in Tweets*, the organizers provide an array of tasks where systems have to automatically determine the intensity of emotions (anger, fear, joy, and sadness) and the intensity of the sentiment (aka valence) of the tweeters from their tweets. They provide annotated datasets for each task with English, Arabic, and Spanish tweets (Mohammad et al., 2018). We define the tasks below:

EI-reg (an emotion intensity regression task): Given a tweet and an emotion E, determine the intensity of E that best represents the mental state of the tweeter with a real-valued score between 0 (least E) and 1 (most E).

EI-oc (an emotion intensity ordinal classification task): Given a tweet and an emotion E, classify the tweet into one of four ordinal classes of intensity of E that best represents the mental state of the tweeter.

V-reg (a sentiment intensity regression task): Given a tweet, determine the intensity of sentiment or valence V that best represents the mental state of the tweeter with a real-valued score between 0 (most negative) and 1 (most positive).

V-oc (a sentiment analysis, ordinal classification, task): Given a tweet, classify it into one of seven ordinal classes, corresponding to various levels of positive and negative sentiment intensity, that best represents the mental state of the tweeter.

We proposed one system to solve the intensity regression tasks (i.e. EI-reg and V-reg) and use it as a feature extractor to train Decision Trees to solve the ordinal classification tasks (i.e. EI-oc and V-oc). We developed two versions of the proposed system for the English and the Arabic language tweets.

Our system is an ensemble of two different approaches. The first one, called N-Channels ConvNet, is a deep learning approach where the second one is an XGboost regressor based on a set of embedding and lexicons-based features.

The rest of the paper is organized as follows:

Proceedings of the 12th International Workshop on Semantic Evaluation (SemEval-2018), pages 193–199
New Orleans, Louisiana, June 5–6, 2018. ©2018 Association for Computational Linguistics

Section 2 presents the tools and the resources that are used. Section 3 describes the proposed system. In Section 4 we report the experimental results, whereas in Section 5 the conclusions and the future work are presented.

2 Resources

This section explains the tools and the resources that have been used in our system.

2.1 Sentiment Lexicons

We used the following lexicons for the English version of our system:

AFINN (Nielsen, 2011), General Inquirer (Stone et al., 1968), Bing-Liu opinion lexicon (HL) (Hu and Liu, 2004), MPQA (Choi and Wiebe, 2014), NRC hashtag sentiment lexicon (Mohammad et al., 2013), NRC emotion lexicon (EmoLex), NRC affect intensity lexicon, NRC hashtag emotion lexicon and Vader lexicon. More details about each lexicon, such as how it was created, the polarity score for each term, and the statistical distribution of the lexicon, can be found in (Jabreel and Moreno, 2016).

For the Arabic version we used the following lexicons:

Arabic Hashtag lexicon, Dialectal Arabic Hashtag lexicon, Arabic Bing Liu lexicon, Arabic Sentiment140 lexicon and Arabic translation of the NRC emotion lexicon. The first two were created manually, whereas the rest were translated to Arabic from the English version using Google Translator. (Mohammad et al., 2016).

2.2 Embeddings

Word embeddings are an approach for distributional semantics which represents words as vectors of real numbers. Such representation has useful clustering properties, since the words that are semantically and syntactically related are represented by similar vectors (Mikolov et al., 2013). For example, the words ”coffee” and ”tea” will be very close in the created space.

We used two publicly available pre-trained embedding models in the English version of our system. The first one was used in (Rouvier and Favre, 2016). It was trained using word2vec (skipgram model) on an unannotated corpus of 20 million English tweets containing at least one emoticon. The second one was provided by (Baziotis et al., 2017). It was trained on a big dataset of 330M English Twitter messages, gathered from 12/2012 to 07/2016 and a vocabulary size of 660K words using Glove algorithm.

Additionally, we have trained two embedding models on 60M English tweets(30M contain positive emoticons, 30M negative ones). The first one was trained by applying word2vec skipgram of window size 5 and filtering words that occur less than 4 times. The dimensionality of the vector was set to 300. The second one was trained using fastText (Bojanowski et al., 2016). The dimensionality of the vector was set to 300.

Similarly, we used two publicly available pre-trained embedding models in the Arabic version of our system and trained two. The first one is the model Arabic-SKIP-G300, provided by (Zahran et al., 2015). Arabic-SKIP-G300 was trained on a large corpus of Arabic text collected from different sources such as Arabic Wikipedia, Arabic Gigaword Corpus, Ksucorpus, King Saud University Corpus, Microsoft crawled Arabic Corpus, etc. It contains 300-dimensional vectors for 6M words and phrases. The second one is Twitter-SG-AraVec (Soliman et al., 2017), which was trained using word2vec skipgram algorithm on 66M Arabic tweets and 1B tokens. The dimensionality of the vector was set to 300.

Our embedding models were trained on the distant supervision corpus (about 16M Arabic tweets) provided by the organizers. We were able to find about 12M tweets. Again, similar to our English embeddings, we trained the two Arabic embedding models.

3 System Description

This section explains the proposed system, whose architecture is shown in Figure 1. First, we preprocess the tweets (Subsection 3.1). Afterwards, we pass them to the N-Channels ConvNet and the XGboost regressors (Subsections 3.2 and 3.3). Finally we ensemble the output of the two systems to get the final result as described in subsection 3.4. The proposed system is also used as feature extractor to train an ordinal Decision Tree classifier. as described in subsection 3.5.

3.1 Preprocessing

Some standard pre-processing methods were applied on the tweets:

- *Normalization*: Each tweet in English was converted to the lowercase. URLs and user-

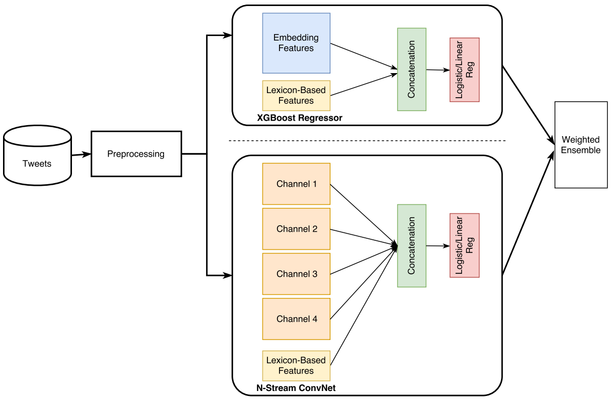

Figure 1: System Architecture.

names were omitted. Non-Arabic letters were removed from each tweet in the Arabic-language sets. Words with repeated letters (i.e. elongated) are corrected.

- *Tokenization*: All English-language tweets were tokenized using Ark Tweet NLP (Gimpel et al., 2011), while all Arabic-language tweets were tokenized using Stanford Tokenizer (Green and Manning, 2010).

3.2 N-Channels ConvNet

Convolutional Neural Networks (ConvNets) have achieved remarkable results in computer vision and speech recognition tasks in recent years. The next subsection explains the architecture of our proposed ConvNet.

3.2.1 Architecture

The N-Channels ConvNet model architecture, shown in the bottom box in figure 1, is inspired by Inception-Net (Szegedy et al., 2016) and the CNN proposed by (Kim, 2014). It is composed of multiple channels followed by a logistic regressor. Figure 2 shows the channel architecture. The input to each channel is a sequence of words $w_1, w_2, ...w_n$ where n is the number of words. We pass the input through an embedding layer to map each word w_i into a real-valued vector. Each channel has its own embedding layer which is initialized by a specific pre-trained embedding model. We use five channels with the four pre-trained embedding models described in subsection 2.2 and a character based one. The result from the embedding layer is a matrix $n \times d_c$ where d_c is the vector dimension. This matrix is passed to a projection or pre-activation layer. The projection layer is nothing but a fully-connected or dense layer whereas the pre-activation layer can be any non-linear function such rectified linear unit (ReLU). Afterward, we feed the projected matrix to three Conv1D. Each one has a different kernel (1, 2, and 3) and 200 filters. To get more details about the architecture of this Conv1D please check (Kim, 2014). We pass the output of each Conv1D through a global max-pooling layer which produces a vector with dimensionality of 200. Finally, the three vectors are concatenated. This yields a vector with dimensionality of 600 that represents the tweet (i.e. the input sequence of words).

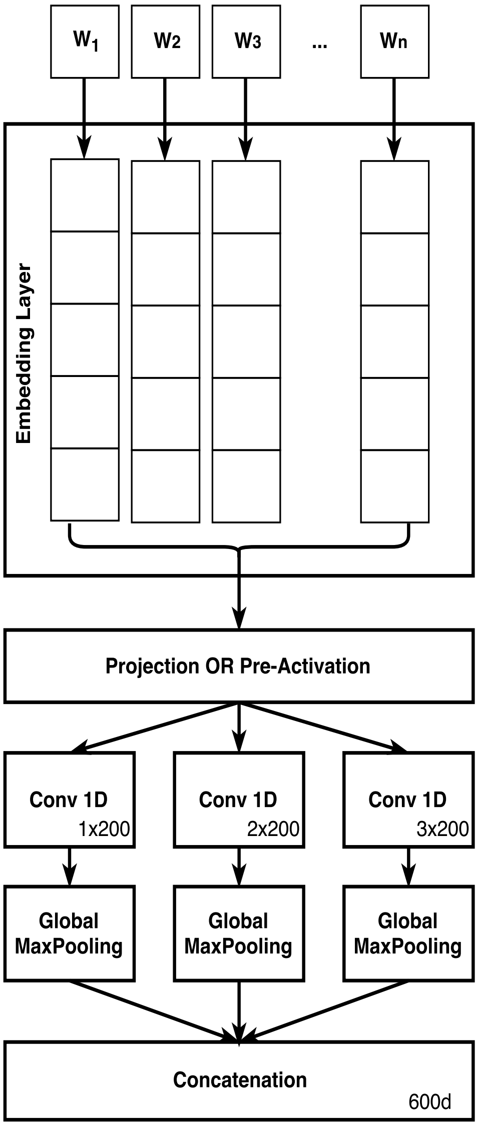

Figure 2: Channel Architecture.

Finally, the outputs of all channels are concatenated with a lexicon-based vector (see next section) and fed to a single sigmoid neuron which gives the intensity of the emotion/valence.

3.2.2 Training

The proposed model was trained by minimizing the mean squared error between the real and predicted intensities. The optimization was done by applying back-propagation through layers via minibatch gradient descent. The training parameters were the following: batch size of 32, 100 epochs and Adam optimization method with learning rate of 0.001, β_1 = 0.9 and β_2 = 0.999 and ϵ = 10^-9. To prevent the over-fitting, we used dropout and early stopping methods.

3.3 XGBoost Regressor

XGBoost (Chen and Guestrin, 2016) has become a widely used and really popular tool among Data Scientists in industry, as it shows great performance on large-scale problems. It is a highly flexible and versatile tool that can work through most regression, classification and ranking problems as well as user-built objective functions.

We trained an XGBoost regressor to give the intensity of the emotion/valence based on the two types of features explained in the next subsection.

3.3.1 Features

Each tweet is represented with a vector by concatenating the following two feature vectors:

Lexicon Features: For each lexicon, we used the sum of the scores provided by the lexicon for each word in the tweet. Let L denote the set of lexicons and $f_i^l(w)$ the score of the word w based on the feature i in the lexicon l (note that some lexicons have only one feature like the sentiment score and some of them have multiple features like anger emotion score, positive score, etc). Then, the set of features that represent a given tweet T and a lexicon $l \in L$ can be obtained as follows:

$$V_{T,l} = \forall_{f_i^l \in F_l} \sum_{w \in T} f_i^l(w) \qquad (1)$$

Here, F_l denotes the set of features in lexicon l.

Embedding Features: We used the *sum* pooling function to obtain the tweet representation in the embedding space. More formally, let us consider an embedding matrix $E \in \mathbb{R}^{d \times |V|}$ and a tweet $T = w_1, w_2, ..., w_n$, where d is the dimension size, $|V|$ is the length of the vocabulary (i.e. the number of words in the embedding model), w_i is i-th the word in the tweet and n is the number of words. First, each word w_i is substituted by the corresponding vector v_i^j in the matrix E where j is the index of the word w_i in the vocabulary. This step ends with the matrix $W \in \mathbb{R}^{d \times n}$. The vector $V_{T,E}$ that represents the tweet T is computed by aggregating the matrix W. This aggregation is done by taking the summation over its columns. The sum spooling function is an element-wise function, and it converts texts with various lengths into a fixed-length vector allowing to capture the information throughout the entire text.

3.3.2 Training

The XGBoost regressor has some parameters that need to be tuned. Table 1 shows the values of each parameter we chose for the different emotions. All those values were chosen using the grid-search on the development sets.

	P	# Est.	S	M	O
Eng.	Anger	300	0.75	5	Logistic
	Fear	300	0.75	5	Linear
	Sadness	300	0.75	5	Logistic
	Joy	300	0.75	7	Linear
	Valence	300	0.75	5	Linear
Ara.	Anger	200	0.9	9	Logistic
	Fear	200	0.9	5	Logistic
	Sadness	200	0.9	5	Logistic
	Joy	200	0.9	5	Logistic
	Valence	200	0.9	9	Logistic

Table 1: The XGBoost regressors parameters. #Est. refers to the number of estimators, S is the subsample, M is the maximum depth and O refers to the objective function.

3.4 Ensemble

We combined the results of the two systems described above with the intention of improving the performance and increasing the generalizability of the final system. We used the weighted average method to achieve that. Let r_1 and r_2 respectively denote the output of the XGBoost regressor and the N-Channels ConvNet system. The final output r was obtained as follows:

$$r = \alpha * r_1 + (1 - \alpha) * r_2; \quad \alpha \in [0, 1] \qquad (2)$$

Table 2 shows the value of α for each individual model. All these values were obtained by grid search on the development set.

3.5 Decision Tree for Ordinal Classification Tasks

To solve the problem of ordinal classification we simply used the proposed model as feature extrac-

	Emotion	α
Eng.	Anger	0.3
	Fear	0.5
	Sadness	0.6
	Joy	0.2
	Valence	0.6
Ara.	Anger	0.5
	Fear	0.0
	Sadness	0.5
	Joy	0.4
	Valence	0.7

Table 2: The value of α for each individual model.

tor and trained a Decision Tree. The idea is to use the emotion/intensity as input feature and use rules generated from the Decision Tree to get the appropriate class. Figure 3 shows as an example the Decision Tree classifier of the $fear$ emotion.

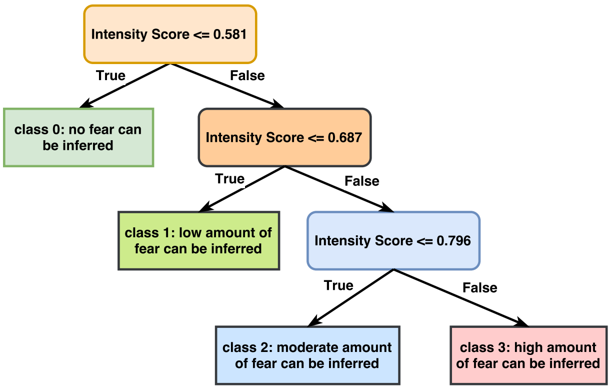
Intensity Score <= 0.581
True
False
class 0: no fear can be inferred
Intensity Score <= 0.687
True
False
class 1: low amount of fear can be inferred
Intensity Score <= 0.796
True
False
class 2: moderate amount of fear can be inferred
class 3: high amount of fear can be inferred

Figure 3: An example of a decision tree classifier.

4 Results

We trained and validated our models on the training and validation sets provided by the organizers. More details about the data and the evaluation metrics can be found in (Mohammad et al., 2018; Mohammad and Kiritchenko, 2018).

Tables 3 and 4 show the results of the emotion and valence intensity regression tasks of our two systems and their combination (the ensemble model). It also shows the baseline results. The evaluation metrics are the Pearson correlation for all samples and for a subset of the test set that includes only those tweets with intensity score greater or equal to 0.5. The values in the tables show the superiority of the N-Channels ConvNet over the XGBoost regressor. For instance, the results of the English version of the emotion intensity task show that the N-Channels ConvNet outperforms the XGBoost regressor by 5.9% with respect to macro-avg measure. The performance of N-Channels Convnet is very close to the ensemble model. The improvement is only 1.2%. The improvement in the final system of the Arabic version is very small (0.3%). The results of the Pearson correlation of samples whose intensity score is greater or equal to 0.5 show that our system can be used as a classifier. This conclusion is confirmed by the results of the ordinal classification tasks, shown in Tables 5 and 6.

As we described in subsection 3.5, our approach to design a system to solve the ordinal classification tasks was to use the intensity score as input feature to train a Decision Tree. During the inference phase we used our system to produce the intensity score for the new (unseen) samples (i.e. use it as feature extractor). Thus, the performance in this phase heavily relies on the performance of the proposed system. This is clearly shown in the results reported in tables 5 and 6. For example, our system gives very good results in the valence intensity regression task for both the English and Arabic versions (the Pearson correlation is 0.828 for both). This affects positively the performance of our system for the valence ordinal classification tasks (the Pearson correlation is about 0.80 for both).

5 Conclusion

We have presented an ensemble model of two different approaches. The first one, called N-Channels ConvNet, is a deep learning approach whereas the second one is an XGBoost regressor based on a set of embedding and lexicons-based features. The ensemble technique helped to improve the performance of the final model in all subtasks. We have realized that The N-Channels ConvNet gives a performance very close to the ensemble model. This observation confirms the fact that deep learning models, and especially ConvNets, have achieved remarkable results in many fields such as computer vision, speech recognition and natural language processing. Distant Supervision is an approach of transfer learning which aims to train a model on a large amount of semi-labeled data and use it as a pre-trained model for training another model on a small amount of fully-labeled data. This approach has been shown to be very efficient. Thus, the authors are considering

		Pearson (all instances)					Pearson5 (gold in 0.5-1)				
		macro-avg	anger	fear	joy	sadness	macro-avg	anger	fear	joy	sadness
Eng.	N-Channels ConvNet	0.712	0.713	0.725	0.718	0.692	0.538	0.575	0.502	0.519	0.555
	XGBoost Regressor	0.653	0.674	0.644	0.625	0.668	0.503	0.563	0.455	0.437	0.555
	Ensemble Model	0.724	0.731	0.733	0.722	0.711	0.560	0.606	0.522	0.525	0.587
	SVM Unigrams	0.520	0.526	0.525	0.575	0.453	0.396	0.455	0.302	0.476	0.350
	Random Baseline	-0.008	-0.018	0.024	-0.058	0.020	-0.048	-0.088	-0.011	-0.032	-0.059
Ara.	N-Channels ConvNet	0,655	0.639	0.628	0.705	0.648	0,516	0.473	0.605	0.465	0.520
	XGBoost Regressor	0.596	0.494	0.540	0.713	0.637	0.464	0.376	0.492	0.449	0.540
	Ensemble Model	0.667	0.627	0.627	0.738	0.675	0.533	0.479	0.604	0.490	0.560
	SVM Unigrams	0.455	0.406	0.435	0.530	0.450	0.353	0.344	0.366	0.332	0.367
	Random Baseline	0.013	-0.006	0.016	-0.010	0.052	-0.007	0.002	0.007	0.011	-0.048

Table 3: EI-reg task results.

		Pearson	Pearson5
Eng.	N-Channels ConvNet	0.825	0.645
	XGBoost Regressor	0.768	0.598
	Ensemble Model	0.828	0.658
	SVM Unigrams	0.585	0.449
	Random Baseline	0.031	0.012
Ara.	N-Channels ConvNet	0.817	0.550
	XGBoost Regressor	0.774	0.571
	Ensemble Model	0.828	0.578
	SVM Unigrams	0.571	0.423
	Random Baseline	-0.052	0.022

Table 4: V-reg task results.

the possibility of using this technique to improve the proposed system.

Acknowledgment

This work was partially supported by URV Research Support Funds (2017PFR-URV-B2-61 and Martí i Franqués PhD grant).

References

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, Vancouver, Canada. Association for Computational Linguistics.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A Scalable Tree Boosting System. *CoRR*, abs/1603.02754.

Yoonjung Choi and Janyce Wiebe. 2014. +/-effectwordnet: Sense-level lexicon acquisition for opinion inference. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1181–1191.

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*, HLT '11, pages 42–47, Stroudsburg, PA, USA. Association for Computational Linguistics.

Spence Green and Christopher D Manning. 2010. Better arabic parsing: Baselines, evaluations, and analysis. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 394–402. Association for Computational Linguistics.

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. ACM.

Mohammed Jabreel and Antonio Moreno. 2016. Sentirich: Sentiment analysis of tweets based on a rich set of features. In *Artificial Intelligence Research and Development - Proceedings of the 19th International Conference of the Catalan Association for Artificial Intelligence, Barcelona, Catalonia, Spain, October 19-21, 2016*, pages 137–146.

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

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.

		Pearson					Kappa				
		macro-avg	anger	fear	joy	sadness	macro-avg	anger	fear	joy	sadness
Eng.	Our system	0.633 (6)	0.651 (5)	0.595 (2)	0.651 (8)	0.636 (6)	0.608 (3)	0.619 (4)	0.574 (3)	0.607 (10)	0.632 (4)
	SVM Unigrams	0.394 (26)	0.382 (27)	0.355 (26)	0.469 (26)	0.370 (29)	0.385 (26)	0.375 (26)	0.331 (25)	0.465 (25)	0.370 (28)
	Random Baseline	-0.016 (37)	-0.062 (38)	0.047 (33)	0.014 (35)	-0.061 (37)	-0.017 (38)	-0.058 (38)	0.035 (32)	0.014 (35)	-0.057 (37)
Ara.	Our system	0.574 (2)	0.572 (1)	0.529 (2)	0.634 (1)	0.563 (3)	0.542 (2)	0.547 (1)	0.516 (2)	0.588 (2)	0.518 (3)
	SVM Unigrams	0.542 (2)	0.315 (6)	0.281 (7)	0.281 (6)	0.396 (6)	0.299 (6)	0.276 (7)	0.249 (6)	0.385 (6)	0.287 (7)
	Random Baseline	0.006 (12)	-0.057 (12)	-0.019 (12)	0.008 (12)	0.092 (11)	0.006 (12)	-0.057 (14)	-0.019 (12)	0.007 (12)	0.091 (10)

Table 5: EI-oc task results.

		Pearson	Kappa
Eng.	Our system	0.796	0.791
	SVM Unigrams	0.509	0.504
	Random Baseline	-0.010	-0.010
Ara.	Ensemble Model	0.809	0.783
	SVM Unigrams	0.471	0.470
	Random Baseline	0.011	0.011

Table 6: V-oc task results.

Saif Mohammad, Mohammad Salameh, and Svetlana Kiritchenko. 2016. Sentiment lexicons for arabic social media. In *Proceedings of 10th edition of the the Language Resources and Evaluation Conference (LREC)*, Portorož, Slovenia.

Saif M. Mohammad. 2016. Sentiment analysis: Detecting valence, emotions, and other affectual states from text. In Herb Meiselman, editor, *Emotion Measurement*. Elsevier.

Saif M. Mohammad and Felipe Bravo-Márquez. 2017. WASSA-2017 shared task on emotion intensity. *CoRR*, abs/1708.03700.

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. 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.

Finn rup Nielsen. 2011. A new anew: Evaluation of a word list for sentiment analysis in microblogs. *CoRR*, abs/1103.2903.

Mickael Rouvier and Benoit Favre. 2016. Sensei-lif at semeval-2016 task 4: Polarity embedding fusion for robust sentiment analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval 2016), San Diego, US*.

Abu Bakr Soliman, Kareem Eissa, and Samhaa R. El-Beltagy. 2017. Aravec: A set of arabic word embedding models for use in arabic nlp. *Procedia Computer Science*, 117:256 – 265. Arabic Computational Linguistics.

Philip Stone, Dexter C Dunphy, Marshall S Smith, and DM Ogilvie. 1968. The general inquirer: A computer approach to content analysis. *Journal of Regional Science*, 8(1):113–116.

Christian Szegedy, Sergey Ioffe, and Vincent Vanhoucke. 2016. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. *CoRR*, abs/1602.07261.

Mohamed A Zahran, Ahmed Magooda, Ashraf Y Mahgoub, Hazem Raafat, Mohsen Rashwan, and Amir Atyia. 2015. Word representations in vector space and their applications for arabic. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 430–443. Springer.