CENTEMENT at SemEval-2018 Task 1: Classification of Tweets using Multiple Thresholds with Self-correction and Weighted Conditional Probabilities

Tariq Ahmad
School of Computer Science
University of Manchester
Oxford Road, Manchester
M13 9PL, U.K.
tariq.ahmad@postgrad.manchester.ac.uk

Allan Ramsay
School of Computer Science
University of Manchester
Oxford Road, Manchester
M13 9PL, U.K.
allan.ramsay@manchester.ac.uk

Hanady Ahmed
CAS, Arabic Department
Qatar University
2713, Al Hala St
Doha, Qatar
hanadyma@qu.edu.qa

Abstract

In this paper we present our contribution to SemEval-2018, a classifier for classifying multi-label emotions of Arabic and English tweets. We attempted "Affect in Tweets", specifically Task E-c: Detecting Emotions (multi-label classification). Our method is based on preprocessing the tweets and creating word vectors combined with a self correction step to remove noise. We also make use of emotion specific thresholds. The final submission was selected upon the best performance achieved, selected when using a range of thresholds. Our system was evaluated on the Arabic and English datasets provided for the task by the competition organisers, where it ranked 2nd for the Arabic dataset (out of 14 entries) and 12th for the English dataset (out of 35 entries).

1 Introduction

Social network platforms such as Facebook, LinkedIn and Twitter are now at the hub of everything we do. Twitter is one of the most popular social network platforms; as recently as 2013 an incredible 21% of the global internet population used Twitter actively on a monthly basis (globalwebindex, accessed 05/2016). Twitter is used by celebrities, movie stars, politicians, sports stars and everyday people. Every day, millions of users share their opinions about themselves, news, sports, movies and many many other topics. This makes platforms like Twitter rich sources of data for public opinion mining and sentiment analysis (Pak and Paroubek, 2010). However, although these corpora are rich, they are somewhat noisy because tweets can be informal, misspelt and contain slang, emoticons (Albogamy and Ramsay, 2015) and made-up words. Furthermore, Arabic tweets have the added complication of dialects in which the same words or expressions can have different connotations.

Multi-label classification of tweets is a classification problem where tweets are assigned to two or more classes. It is considered more complex than traditional classification tasks because the classifier has to predict several classes.

There has been much work in the areas of sentiment detection (Rosenthal et al., 2017), emotion intensity (Mohammad and Bravo-Marquez, 2017) and emotion categorisation (Hasan et al., 2014). Sentiment analysis aims to classify tweets into positive, negative, and neutral categories, emotion intensity is determining the intensity or degree of an emotion felt by the speaker and emotion categorisation is the classification of tweets based on their emotions. The most commonly used classification techniques are Naive Bayes and Support Vector Machines (SVM). Some researchers report that SVMs (Barbosa and Feng, 2010) perform better while others support Naive Bayes (Pak and Paroubek, 2010). Furthermore, sophisticated techniques such as deep neural networks have also been proposed but such techniques are rarely used by non-experts of machine learning in practice (Sarker and Gonzalez, 2017) and they also take a long time to train.

We propose a simple and effective method to classify tweets that performs reasonably well. Our system does not make use of any lexicons or stop word lists and is quick to train.

2 Methods

The SemEval Task E-c requires the classification of tweets into either a neutral emotion or one of eleven emotions (Mohammad et al., 2018). Datasets for tweets are made available in three languages; Arabic, English and Spanish. We focus firstly on Arabic and then English because this links well with our existing work. Datasets from previous SemEval tasks are also available if

Proceedings of the 12th International Workshop on Semantic Evaluation (SemEval-2018), pages 200–204
New Orleans, Louisiana, June 5–6, 2018.

required. We use the SemEval-2018 development and training data for training our system, with no external resources such as sentiment dictionaries or other corpora. We use the training set to compute scores for each of the classes in conjunction with a self correction stage and a multi-threshold stage to obtain an optimal set of scores. Apart from the preprocessing steps, notably stemming, we use exactly the same machinery for the two languages. We now briefly discuss our approach.

Preprocessing. Tweets are preprocessed by lowercasing (English tweets only), identifying and replacing emojis with emojis identifiers, tokenising and then stemming. We developed two tokenisers; one that is NLTK based and does not preserve hashtags, emoticons, punctuation and other content and one that is "tweet-friendly" because it preserves these items. Emojis cause us technical problems due to their surrogate-pair nature so we replace emojis with emoji identifiers (e.g. _45_). We also separate out contiguous emojis because we want, for example, the individual emojis in a group of repeating unhappy face emojis to be recognised, and processed, as being the same emoji as a single unhappy face emoji. We remove usernames because we believe they are noise since, by and large, they will not reappear in the test set, are not helpful to us and if not removed will compromise our ability to detect useful information. Arabic tweets are stemmed using a stemmer developed specifically for Arabic tweets by Albogamy and Ramsay (Albogamy and Ramsay, 2016). English tweets are stemmed by taking the shortest result from Morphy (Fellbaum, 1998) when tokens are stemmed as nouns, verbs, adjectives and adverbs. Although there are surprisingly few examples of these, we believe that multi-word hashtags, joined by underscore or a dash, also contain useful information so we leave the hashtag as is but also take a copy of the hashtag and split it into its constituent words. This is so that where possible we improve the quality of information in the tweet. Stop word lists are not used at any stage. We debated using stop words vs insignificant words and, as in our previous work (Ahmad and Ramsay, 2016), we prefer to let our algorithms exclude these words. We do however remove less common words on the grounds that if they do not appear very often then we are unlikely to learn anything from them. The English training dataset contains approximately 6300 distinct words after preprocessing, we find that taking the top 2500 of these gives us the most common words and the best results.

Our approach is not to collect scores for individual emotions, instead we collect scores relative to the other emotions. Constructing scores in this manner allows us to observe that words such as "blessed" are much more significant for emotions such as "joy", "love" and "optimism" than they are for "anger" and "anticipation". Words that are insignificant will have small scores, words that are significant will have large scores and by using a varying threshold we can determine a best set.

Base set. Every tweet in the training dataset is tokenised and we count how many tweets each token in the tweet occurs in. We also remove singletons and calculate an IDF for each token. We iterate through the tokens for each tweet to create a base set of scores and obtain a count of how many times each token occurs in each of the 11 emotions as well as a count of the total number of tokens in each emotion. In a later stage we iterate over a range of thresholds, this base set is the starting point in each and is modified by the various processes as described below.

Conditional probabilities. We now use this base set to create a set of emotion probabilities for each token. One, common, way of using probabilities is in conjunction with Bayes Theorem. However, this does not seem to work very well for this task hence we perform the following steps. We calculate the probability of each token appearing in each emotion using P(T|E). We do this only on the top 2500 most important tokens in the dataset, i.e. those with the highest IDF scores. We normalise these probabilities by dividing each value by the sum of all the probabilities for this token for all emotions. We get an average value for these values and subtract this from each of the scores to calculate the distance from the mean. This is, essentially, a local IDF step to ensure that if a token is equally common for all emotions then we do not allow it to contribute to any of them, and if it is below the overall average for a emotion we want it to be allowed to vote against it.

We want to assign extra weight to tokens that have very skewed distributions, hence we multiply each score by the standard deviation. This empha-

sises the contribution of such tokens to the emotion and allows us to remove unhelpful tokens. In this way we create a set of emotion scores for each token for every emotion.

Self-correction. We want to remove tokens that we have incorrectly assigned to emotions. We classify each tweet to determine which emotions it demonstrates and we identify the tokens that led us to these conclusions. A tweet is classified for each emotion by adding the scores for each token for each emotion. These scores are normalised and compared to a threshold t. If the value is less than t we deduce the tweet did not demonstrate the emotion, otherwise it did demonstrate the emotion. We are unsure what a good threshold is so we use a range of values for t from 0 to 1 (in steps of 0.1) to create score sets. We calculate the Jaccard for each of these and use the best one of these for classification. This approach is based on Brills (Brill, 1995) suggestion that one should attempt to learn from ones own mistakes.

As each tweet is classified we compare our prediction to the gold standard. For the ones that we predict correctly we increment a counter for each token against the correctly classified emotion. Similarly, for the ones where we failed to classify the tweet correctly we decrement the counter for each token against the incorrectly classified emotion. When all tweets have been classified we examine these counters. For each token, if we have an overall negative score for an emotion we deduce that the token is unhelpful in classifying tweets for that emotion and we downplay its significance in further calculations. Using this technique we are able to remove tokens such as "terrifying" from contributing to emotions such as "love". We have tried repeating this process multiple times, but we find that beyond one iteration the improvement is insignificant. A possible explanation for this may be because the actual numbers of tokens that are removed are quite small; 1% for Arabic and 5% for English.

Per-emotion thresholds. The raw data for each emotion is different and, hence, we find that a single fixed threshold across all emotions produces poor results. We therefore try a range of thresholds from 0 to 1 in increments of 0.1 to classify tweets, using the same mechanism described above, but this time on an emotion-by-emotion basis to generate an individual threshold for each emotion.

SemEval results. We classify the training data using our sets of scores and per-emotion thresholds. We identify the set with the best Jaccard score and use it to classify the test data to generate our eventual submission file.

2.1 Other Strategies

Increased training data. We believe that having more training data might improve our classifer. One of the obvious places to get more data is from the datasets for some of the other tasks, specifically EI-reg and EI-oc. A key problem with this data is that both of these tasks only supply datasets for anger, fear, joy and sadness. The El-reg dataset is marked up with a per-tweet intensity value between 0-1 that represents the mental state of the tweeter. The EI-oc dataset tweets are marked up with one of four ordinal classes (0,1,2,3). To expand our training dataset we extract tweets with values of 0.5 and above from the El-reg datasets and tweets with a value of 3 from the EI-oc dataset. The best Jaccard score we obtain with this expanded dataset is 0.417 (English). When we extract tweets with values of 0.9 or above from the EI-reg dataset we improve the quality of tweets, at the cost of decreasing the number of tweets extracted, and this slightly improves our Jaccard to 0.429.

Similarly, the competition organisers also make available a corpus of 100 million English tweet IDs. We download 10,000 of these filtered on words that we believe are representative of the emotions we are looking for e.g. "angry", "elated", "trusting". A serious weakness with this technique, however, is that the accuracy of this data is compromised, we therefore classify this data using our classifier. We then combine this data with the standard English dataset and classify it again. We do not want this data to be more relevant than the real data, so we weight down the scores from this data. The best Jaccard score we obtain with this expanded dataset is 0.430.

Latent semantic analysis (LSA). Latent Semantic Analysis (LSA) is a theory and method for extracting and representing the contextual-usage meaning of words by statistical computations applied to a large corpus of text (Landauer and

Tweet tokeniser	Split hashtags	Stem	Tune	Multi-threshold	Jaccard (AR)	Jaccard (EN)
					0.324	0.340
✓					0.318	0.340
✓	✓				0.333	0.349
✓	✓	✓			0.342	0.401
✓	✓	✓	✓		0.370	0.431
✓	✓	✓	✓	✓	**0.452**	**0.455**

Table 1: Results.

Dumais, 1997). Essentially, to improve our classifier we need to improve the quality of our tweets. We use LSA to find words in tweets that are similar to other words, e.g. "car" and "automobile". We do not have the computing power to do this on a per-tweet basis so we do this on a per-emotion basis. The concepts we find, however, are not very reliable, e.g. "blessed" and "happiness". We expand our tweets with these words but find that this does not improve our scores. A possible explanation for this might be because of the relatively small numbers of tweets in the datasets.

Duplicate tweets. We note that there are tweets in the English dataset that are semantically similar, e.g. *"You offend me, @Tansorma"* and *"@SunandBeachBum 'you people' infuriate me!"*. It may be possible to use *clustering* (Sarker and Gonzalez, 2017) to relate tweets like these as a means to removing duplicates. We further note that there are many cases of tweets that differ only by hashtags or emojis, e.g. *"@britishairways term 5 security queues at arrivals"* and *"@britishairways term 5 security queues at arrivals #shocking"*. A further study could assess the impact of using Minimum Edit Distance (Wagner and Fischer, 1974) on this later data to improve the quality of the dataset.

Emoticon weighting. Emoticons have proved crucial in the automated emotion classification of informal texts (Novak et al., 2015). To increase their significance we double their raw count values. We find that this increases the accuracy of our classifier by 0.44% for both Arabic and English.

Word frequencies. We try to use the word frequency as an extra weight to further dampen the contribution of words that are low frequency because low frequency words do not contribute very much. However, because we have earlier taken only the 2500 commonest words we find that this does not improve our scores.

2.2 Computing Resources

The system was written in Python on a MacBook Pro, 2.7 GHz Intel Core i5, 8 GB RAM. The training and classification phase takes approximately 15 minutes.

3 Results, Comments and Conclusion

We described a self-correcting, multi-threshold, classifier to solve the problem of multi-label classification of tweets.

We find that due to the nature of the data it is difficult to accurately distinguish between emotions such as "joy" and "love" because many of the words that score highly for "joy " also score highly for "love", e.g. "rejoice", "birthday" and "cheerful". Consequently when a tweet is labelled as "love" it is highly likely that it will also be labelled as "joy". We find similar issues with "anger" and "disgust", although not to the same extent, because words like "shit" and "hate" score highly for both emotions. Overall, we believe that we score much higher on emotions such as "anger", "joy", "love" and "disgust", than on "trust" "anticipation", "optimism" and "pessimism".

Our results, given in Table 1, show that although processes such as lowercasing, tokenising and stemming do contribute, the tuning stage and the introduction of multiple thresholds yield the biggest improvements. This is because removing words which are implicit in the classifier making wrong decisions and allowing each emotion to have its own threshold are obviously sensible things to do.

One unanticipated finding was that our tweet-friendly tokeniser has an adverse effect decreasing the Jaccard score when it is used. A possible ex-

planation for this is that the simple tokeniser removes # and @ symbols, thus modifying hashtags such as "#sleep" into "sleep" and allowing them to combine with the word "sleep" in other tweets. On the other hand the tweet-friendly tokeniser preserves the "#sleep" hashtag and it therefore cannot combine with the word "sleep". We want the best of both worlds so we preserve our hashtag but also take a copy and split it into its constituent words.

Contrary to expectations, the performance improvement gained from using our Arabic stemmer is disappointingly low at just 2.67%. We believed that our Arabic stemmer would have a bigger impact than demonstrated because the stemmer is aimed at, and specifically developed for, Arabic tweets. In fact our simplistic Morphy English stemmer produced a better improvement of 14.8% for English than our carefully tuned Arabic stemmer did for Arabic.

The scores we achieved put us 2nd for the Arabic dataset and 12th for the English dataset despite the fact that we use no external resources, we simply train on the basis of the SemEval data. We will be carrying out further experiments to see whether adding external resources would give us further improvement.

Acknowledgments

This publication was made possible by the NPRP award [NPRP 7-1334-6-039 PR3] from the Qatar National Research Fund (a member of The Qatar Foundation). The statements made herein are solely the responsibility of the author[s].

References

Apoorv Agarwal, Boyi Xie, Ilia Vovsha, Owen Rambow, and Rebecca Passonneau. 2011. Sentiment analysis of twitter data. In *Proceedings of the workshop on languages in social media*, pages 30–38. Association for Computational Linguistics.

Tariq Ahmad and Allan Ramsay. 2016. Linking tweets to news: Is all news of interest? In *International Conference on Artificial Intelligence: Methodology, Systems, and Applications*, pages 151–161. Springer.

Fahad Albogamy and Allan Ramsay. 2015. POS tagging for arabic tweets. *Recent Advances in Natural Language Processing*, page 1.

Fahad Albogamy and Allan Ramsay. 2016. Unsupervised stemmer for arabic tweets. In *Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT)*, pages 78–84.

Anil Bandhakavi, Nirmalie Wiratunga, Deepak Padmanabhan, and Stewart Massie. 2017. Lexicon based feature extraction for emotion text classification. *Pattern recognition letters*, 93:133–142.

Luciano Barbosa and Junlan Feng. 2010. Robust sentiment detection on twitter from biased and noisy data. In *Proceedings of the 23rd international conference on computational linguistics: posters*, pages 36–44. Association for Computational Linguistics.

Eric Brill. 1995. Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging. *Computational linguistics*, 21(4):543–565.

Christiane Fellbaum. 1998. *WordNet*. Wiley Online Library.

globalwebindex. accessed 05/2016. Twitter now the fastest growing social platform in the world.

Maryam Hasan, Elke Rundensteiner, and Emmanuel Agu. 2014. Emotex: Detecting emotions in twitter messages.

Thomas K Landauer and Susan T Dumais. 1997. A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review*, 104(2):211.

Saif M. Mohammad and Felipe Bravo-Marquez. 2017. Emotion intensities in tweets. In *Proceedings of the sixth joint conference on lexical and computational semantics (*Sem)*, Vancouver, Canada.

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.

Petra Kralj Novak, Jasmina Smailović, Borut Sluban, and Igor Mozetič. 2015. Sentiment of emojis. *PLOS ONE*, 10(12):1–22.

Alexander Pak and Patrick Paroubek. 2010. Twitter as a corpus for sentiment analysis and opinion mining. In *LREc*, volume 10, pages 1320–1326.

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.

Abeed Sarker and Graciela Gonzalez. 2017. HLP @ UPenn at SemEval-2017 Task 4A: A simple, self-optimizing text classification system combining dense and sparse vectors. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 640–643.

Robert A Wagner and Michael J Fischer. 1974. The string-to-string correction problem. *Journal of the ACM (JACM)*, 21(1):168–173.