MSc in Computer Science

Department of Computer Science

Master of Science Thesis

Sentiment Analysis for Tweets

Konstantinos Korovesis

Athens 2018

Acknowledgements

I would like to thank my supervisor Prof. Ion Androutsopoulos for introducing me to the world of NLP and for giving me the opportunity to work on this exciting field. Furthermore, I would like to thank C. Baziotis, J. Pavlopoulos and J. Koutsikakis for their useful insights and assistance. Finally, I would like to thank Irene and my family for their support and understanding.

Abstract

Sentiment Analysis is a field of Natural Language Processing which addresses the problem of extracting sentiment or, more generally, opinion from text. Obtaining deeper insights on that topic can be very valuable for a range of fields such as finance, marketing, politics and business. Previous research has shown how sentiment and public opinion can affect stock markets, product sales, polls as well as public health. This thesis researches the message sentiment polarity classification problem in Twitter aiming to classify messages based on the polarity of the sentiment towards a specific topic, where the tweets and the topics are always given. The dataset analyzed and the evaluation metrics considered are provided by the SemEval 2017 International Workshop and the 4th task about "Sentiment Analysis in Twitter". This task includes five subtasks, two of which were eventually engaged in this research according to the implemented approach. First, subtask B is a binary classification task, where the goal is to classify messages into two classes, positive and negative regarding the sentiment towards the topic. Following, subtask C where the target is to classify messages in a five-scale sentiment polarity from highly negative, negative, neutral, positive to highly positive, based on the sentiment towards a given topic. We re-implemented and experimented with two deep learning models for both subtasks; a Convolutional Neural Network (CNN) and a state-of-the-art Recurrent Neural Network with context attention (Att-BiLSTM+WL). We compare both models to the baselines of the challenge and show that the Att-BiLSTM+WL outperforms the other models, in both subtasks, with all evaluation measures but one.

Περίληψη

Η Ανάλυση Συναισθήματος είναι ένας τομέας της Επεξεργασίας Φυσικής Γλώσσας ο οποίος αντιμετωπίζει το πρόβλημα της εξαγωγής συναισθημάτων ή γενικότερα γνώμης από κείμενο. Η απόκτηση βαθύτερων γνώσεων σχετικά με το θέμα αυτό μπορεί να είναι πολύτιμη για μια σειρά πεδίων όπως τα χρηματοοικονομικά, το μάρκετινγκ, η πολιτική και οι επιχειρήσεις. Προηγούμενες έρευνες έχουν δείξει πώς το συναίσθημα και η κοινή γνώμη μπορούν να επηρεάσουν τις χρηματιστηριακές αγορές, τις πωλήσεις προϊόντων, τις δημοσκοπήσεις καθώς και τη δημόσια υγεία. Αυτή η διπλωματική εργασία ερευνά το πρόβλημα της ταξινόμησης της πολικότητας του συναισθήματος μηνυμάτων στο Twitter με στόχο την ταξινόμησή τους με βάση την πολικότητα του συναισθήματος προς ένα συγκεκριμένο θέμα, όπου δίνονται πάντα τα tweets και τα θέματα. Το σύνολο δεδομένων που αναλύεται και οι μετρήσεις αξιολόγησης που εξετάζονται παρέχονται από το SemEval 2017 International Workshop Task 4 για το "Sentiment Analysis in Twitter". Αυτός ο διαγωνισμός περιλαμβάνει πέντε υπό-εργασίες, δύο από τις οποίες εν τέλει προσεγγίσαμε σε αυτή την έρευνα. Πρώτον, η υπό-εργασία Β είναι μια εργασία δυαδικής ταξινόμησης, όπου ο στόχος είναι να ταξινομηθούν τα μηνύματα σε δύο κατηγορίες, θετικά και αρνητικά σχετικά με το συναίσθημα προς το θέμα. Ακολούθως, η υπό-εργασία Γ όπου ο στόχος είναι να ταξινομηθούν τα μηνύματα, ως προς κάποιο θέμα, σε πέντε τάξεις πολικότητας, πολύ αρνητική, αρνητική, ουδέτερη, θετική και πολύ θετική. Εφαρμόσαμε δύο μοντέλα βαθιάς μάθησης και εκτελέσαμε πειράματα για τις δύο υπό-εργασίες; ένα CNN και ένα τελευταίας τεχνολογίας RNN που περιλαμβάνει και μηχανισμό Attention (Att-BiLSTM+WL). Συγκρίνουμε και τα δύο μοντέλα με τα βασικά συστήματα (baselines) του διαγωνισμού και δείχνουμε ότι το Att-BiLSTM+WL ξεπερνά τα άλλα μοντέλα, και στις δύο υπό-εργασίες, με όλα τα μέτρα αξιολόγησης, πλην ενός.

Content

1. INTRODUCTION 7

1.1 The Sentiment Analysis task 7

1.2 Message Polarity Classification in Twitter 8

1.3 Approaches to Sentiment Classification 9

1.4 The purpose of this work 10

1.5 Thesis Outline 11

2. RELATED WORK 13

3. TOPIC-BASED SENTIMENT ANALYSIS MODELS 15

3.1 Datasets 15

3.2 Preprocessing 16

3.3 Baselines 17

3.3.1 Embedding Layer 18

3.3.2 Convolutional neural network baseline 18

3.3.3 Bidirectional LSTM baseline 20

3.4 Att-BiLSTM+WL model 21

3.4.1 Mean Pooling and Annotation 23

3.4.2 Context Attention Layer 23

3.4.3 Regularization 24

3.4.4 Optimization 24

3.4.5 Hyper-parameters 25

3.5 Class Balancing 26

4. EXPERIMENTS 27

4.1 Evaluation Measures 27

4.2 Experimental Setup 29

4.3 Evaluation at Validation 29

4.4 Evaluation at Test 32

5. CONCLUSIONS AND FUTURE WORK 34

5.1 Conclusions 34

5.2 Future work 34

BIBLIOGRAPHY 35

List of Figures

Figure 1. CNN model with 3 filters. 20
Figure 2. The Att-BiLSTM+WL model: A 2-layer bidirectional LSTM with attention. 22
Figure 3. Att-BiLSTM+WL Accuracy in 30 epochs, subtask B. 30
Figure 4. Validation Accuracy of Att-BiLSTM+WL and CNN model, subtask B. 30
Figure 5. Att-BiLSTM+WL macro-MAE in epochs, subtask C (lower is better). 31
Figure 6. Validation macro-MAE of Att-BiLSTM+WL and CNN model, subtask C (lower is better). 31

Lists of Tables

Table 1. Tweets, annotated for their sentiment polarity. 9
Table 2. Dataset statistics for Task 4 subtask B and C. 16
Table 3. Preprocessing examples. 17
Table 4. Confusion Matrix 27
Table 5. Scores in Subtask B (higher is better), Bx indicates a baseline. 32
Table 6. Scores in Subtask C (lower is better), Bx indicates a baseline. 33
Table 7. The effect of weighted loss (WL) in Subtask B (higher is better). 33
Table 8. The effect of weighted loss (WL) in Subtask C (for MAE lower is better). 33

Notation

NLP	Natural Language Processing
SA	Sentiment Analysis
TSA	Topic-based Sentiment Analysis
OM	Opinion Mining
SM	Statistical Modeling
ML	Machine Learning
SVM	Support Vector Machine
NLTK	Natural Language Toolkit
NN	Neural Network
ANN	Artificial Neural Networks
DNN	Deep Neural Networks
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
LSTM	Long Short Term Memory
SGD	Stochastic Gradient Descent
MAE	Mean Absolute Error
WLF	Weighted Loss Function

1. Introduction

1.1 The Sentiment Analysis task

Sentiment Analysis (SA), a kind of Opinion Mining (OM), is a field of Natural Language Processing (NLP) whose goal is to extract the emotion, sentiment or more generally opinion expressed in a human written text. The text mostly derives from social media, product reviews and blogs. While the term *opinion* or *sentiment* is quite generic, the field of study attains a number of tasks. Some of these are, identifying the stance on a target or topic, for instance "Climate Change", extracting the opinion on a product from a review or detecting sentiment polarity in a message. Sentiment Analysis is performed on various linguistic levels. The standard ones are document level, sentence level and aspect or entity level (Appel et al. 2016). Sentiment Analysis has plenty of applications in business, marketing and politics. Determining people's opinion is key for future planning in many fields**.** SA can be used to evaluate future business plans based on public opinion on a new product. Trends in product sales can be pre-identified by measuring the sentiment of the customers. Many marketing agencies propose to companies the right direction for advertising a product based on public sentiment, which is extracted from messages in social media or product reviews. In addition, political parties plan their campaigns on public sentiment that can be extracted from text in social networks, blogs and forums.

Sentiment analysis is not an easy project. There are issues that can throw the analysis off and need attention. For instance, tweets can be sarcastic or contain ambiguous words, which often lead to misclassifying the polarity of the tweet. For example:

- "Shut up. And take my money."

which actually refers to a "must buy" product although the message could be classified as negative because it contains negatively charged words. In addition, in a product review there might be a case of a text such as the following, which expresses both a positive and a negative opinion:

- "*The tuna was cooked perfectly but the miso dressing wasn't tasty at all*".

Also, in Twitter, as well as in product reviews, the polarity if often unclear, for example:

- "*Saakashvili is pushing his own agenda here. The Ukrainian economy is growing, although corruption is still a problem*"

- "*Although some vaccines protect our children, they still have potential to be very toxic*".

A Sentiment Analysis system should also handle negation (i.e., "not good"); perform some kind of word sense disambiguation; and in the case of multiple sentiments and sentiment targets, be able to classify them accordingly. If a message has negative sentiment towards a topic, while expressing positive sentiment towards another topic, then the system should classify the message for each topic accordingly. Such a system should also be able to accurately detect irony which is challenging and part of ongoing research.

1.2 Message Polarity Classification in Twitter

Twitter is a social medium, micro-blogging site where users can post text messages, commonly referred as *tweets*[1]. The number of monthly active Twitter users in the fourth quarter of 2017 was 300 million[2] while approximately 90% of the tweets are public and can be collected for research without violating user anonymity. Tweets are available in real-time, through Twitter's streaming channel API[3]. Tweets can be filtered both by time and location that they were published.

Messages in Twitter usually include emoticons, misspellings (e.g., "*Comeee onnnnn fineee*, *waaaay too")*, slang language (e.g., "*hooked on",* for being addicted to something, "*sick",* for something very good*)* in addition to normal text. Tweets normally include hashtags which many times indicate the topics of the message (e.g., #Yemen, #thankyouobama, #BlackLivesMatter). These deviations in text should be handled or used to gain information regarding sentiment. In Table 1 you can see a few examples of messages in Twitter and their annotated label of sentiment. All annotations were performed on CrowdFlower.[4]

For the 4th SemEval task of "Sentiment Analysis in Twitter" messages should get classified based on the polarity of the opinion expressed in the tweet. The goal of this thesis is to study and examine Topic-based Message Polarity Classification which is described in subtasks B and C of Task 4 in SemEval 2017 (Rosenthal, Farra, and Nakov 2017). The goal of these two subtasks is to classify the sentiment of a tweet, towards a predefined topic. Subtask B employs a two-point

[1] https://twitter.com
[2] https://identitygroup.co.uk/digital-marketing-2016/statistic_id282087_twitter_-number-of-monthly-active-users-2010-2015/
[3] https://developer.twitter.com/en/docs/tutorials/consuming-streaming-data.html
[4] https://www.figure-eight.com/platform

scale; positive or negative, assuming that there is no neutral class. By contrast, subtask C employs a more elaborate five-point scale; i.e., highly negative, negative, neutral, positive and highly positive.

The task can be addressed in many different ways, such as using sentiment lexicons, using humans, using ontologies, etc. For the purpose of this thesis we used Machine Learning (ML) techniques to address the problem. We employ a supervised learning model, which gets trained to label messages based on their expressed sentiment polarity. The model is then tested and evaluated for its performance on accuracy and other metrics.

Positive	Negative
#Greece #Israel #Islam #Jerusalem Greek pilot: We are your friends, and we are always here for you.	POTUS You mean all the ones you created through your illegal wars & genocide across the Middle East?#Libya #Syria #Yemen
Remember; Clinton WON the popular vote by MILLIONS and probably won electoral vote too (if not for Russian intervention)#AuditTheVote	The US public essentially decided an election based on amplified lies and suppressed truths--and some people want a direct popular vote!
Tesla Model S P100D so much more than cutting edge technology #tesla #models #model3 #elonmusk	Tesla's autopilot model highlights the dangers of self-driving cars and potential flaws in engineering.

Table 1. Tweets, annotated for their sentiment polarity.

1.3 Approaches to Sentiment Classification

The existing methods for Sentiment Analysis can be grouped into two main categories.

1. Knowledge-Based
2. Machine Learning

In knowledge-based methods, also called Lexicon-based sentiment classification, the target is to construct or use existing sentiment word lexicons with indicated sentiment labels for the words or phrases in the text. The classification of the text is defined by rules; e.g., a function over the

words, such as the sum of word polarities (Taboada et al. 2011). This approach does not require any training (other than forming a lexicon, if required). However, it requires powerful linguistic resources[5] to extract knowledge from words, which are not always available.

Hu and Liu (2004a) built a lexicon, using only WordNet and a list of labeled seed adjectives. This list contains only positive adjectives (e.g., *great, amazing, nice, cool)* and negative adjectives (e.g., *bad, boring)*. Their method retrieves and automatically labels the synonyms (same polarity) and antonyms (opposite polarity). This process allows the list to grow into a lexicon. A drawback of this approach is that it is only applicable in languages where WordNet is available. In any case, the knowledge-based method may be difficult due to noise in text data, while manually creating rules to combine information about words obtained from the sentiment lexicons takes time and effort.

On the other hand, Machine Learning requires training a model to predict the polarity of the text. The model is trained with text messages, labeled for their sentiment and represented as feature vectors. The latter conventionally requires text preprocessing using language processing tools like NLTK[6]. Text preprocessing mainly involves tokenization, stemming, tagging, and possibly parsing of the text. The selection of the appropriate features from data is crucial and has proven to be a major issue and is always a key objective for researchers.

Previous work on sentiment analysis has exploited well-known supervised machine learning methods, such as Naive Bayes (Martinez-Arroyo and Sucar 2006) , SVMs (Vinet and Zhedanov 2010), Random Forests (Ho 1995), (Wahid et al. 2017). More recent work uses deep learning models (Goldberg and Hirst 2017), especially Recurrent Neural Networks (RNN) and Convolutional Neural Network (CNN). This thesis also focuses on deep learning models.

1.4 The purpose of this work

The purpose of this thesis is to study, build and evaluate a Topic-based Sentiment Analysis (TSA) model, re-implementing a state-of-the-art deep learning neural network capable of classifying messages from Twitter in respect to the sentiment polarity of the message towards a given topic. By using SemEval 2017 Task 4 datasets and evaluation metrics we created a TSA model based on a top scoring paper (Baziotis, Pelekis, and Doulkeridis 2017). We named our model Att-

[5] https://nlp.stanford.edu/links/linguistics.html
[6] http://www.nltk.org/

BiLSTM+WL (Bidirectional LSTM with Attention mechanism and weighted loss). Our implementation of the system of Baziotis et al. outperforms a strong baseline based on a Convolutional Neural Network (CNN) model (Kim 2014) and the baselines of the competition in binary sentiment polarity and five-scale sentiment polarity classification. We compare the results of these two different neural networks for this task, the bidirectional LSTM with an attention mechanism (Bi-LSTM) and the CNN model. These types of neural networks have the ability to understand and recognize patterns in text data and perform very well in text classification (Kim 2014; Tang, Qin, and Liu 2015; Vu et al. 2016).

1.5 Thesis Outline

The structure of the remainder of this thesis is the following:

In Chapter 2 we present previous work in the field of SA in social media, blogs and webpages in general. We show how important it is to understand and identify public opinion, which impacts brand marketing, product sales as well as stock markets. We also present related work on SA for polling political parties and how it can predict the sentiment and the opinion of voters accurately by contrast to standard methods. In addition, we cite more work in sentiment analysis for detecting major events, identifying public health issues and mapping people's demography and health characteristics and the "geography of happiness", a term describing the correlation between sentiment and place.

Our main goal in Chapter 3 is to present our work on message polarity classification in Twitter. We provide a full description of the task and the dataset that we train the Att-BiLSTM+WL model on. Furthermore, we describe our strong baselines; a bidirectional RNN with LSTM cells and attention; and a CNN-based baseline. Next, we give an in depth analysis of all layers of the Att-BiLSTM+WL deep learning model. Here we also present an analysis of the individual key procedures and techniques that constitute the training process, including regularization, class balancing and training. Finally, we present the CNN model that we built for comparing the final results with the Att-BiLSTM+WL model.

In Chapter 4 we describe the evaluation measures for the SA task and how the experiments were conducted. Also, we explain how we worked during the training process, what choices we took regarding when to stop training and how to validate the model based on the experiments. In the

results we present the scores of the Att-BiLSTM+WL model by contrast to the CNN model and the baselines of SemEval 2017 Task 4.

Finally, Chapter 5 summarizes the work of this thesis and proposes future work.

2. Related Work

Sentiment Analysis has been an attractive object of study for AI researchers, computational linguists, cognitive scientists and neurobiologist. As mentioned in Section 1.3 one of the most successful approaches for Sentiment Analysis is Nature Language Processing with Machine Learning (ML) techniques. The fundamental requirement for using supervised ML to be able to solve classification and regression problems, is data availability. Thus, while more and more text data become available through blogs, websites and social media, where people can publish their opinion on many subjects, including politics, products, events and business among others, more researchers studied SA. The first papers on SA were focused on reviews for movies (Pang, Lee, and Vaithyanathan 2002), (Turney 2001), (Pang et al. 2002), (Pang and Lee 2005) and (Popescu and Etzioni 2005) or products (Hu and Liu 2004b), (Popescu and Etzioni 2005). Following those studies, other focused on the analysis of sales of products such as books, movies and videogames based on customers opinions (Chevalier and Mayzlin 2006), (Mishne and Glance 2006), (Liu et al. 2007), (Zhu and Zhang 2010).

With the rapid growth of social media like Twitter, more attention was drawn towards social media content as in (Jansen et al. 2009), (Asur and Huberman 2010), (Arias, Arratia, and Xuriguera 2013). In addition to Sentiment Analysis and the impact of people's opinion on sales, extensive research has been done on separate fields of finance and economics. As demonstrated by Lemmon and Portniaguina (2006) and Han (2008) there is a correlation between the sentiment and confidence of the investors and the stock market. Moreover Gilbert and Karahalios (2010) show that "estimating emotions from weblogs provides novel information about future stock market prices.", while Bollen, Mao, and Zeng (2011) explored the fact that national events affect people's emotions and the relationship of their emotions to the value of Dow Jones Industrial Average (DJIA). Due to these findings, more work has been done in the last years on the subject (Oh and Sheng 2011; Zhang, Fuehres, and Gloor 2011; Makrehchi, Shah, and Liao 2013; Si et al. 2013; Smailović et al. 2013; Sprenger et al. 2014; Sprenger et al. 2014). Quoting Mitchell et al. (2013) "Companies should pay more attention to the analysis of sentiment related to their brands and products in social media communication as well as in designing advertising content that triggers emotions."

The ability to measure public opinion on social and political affairs is critical for political parties. The usual methods such as polls are expensive, they may not be accurate and the results are not representative of the public sentiment. Overall polls are unreliable. In addition, getting people's opinion by asking questions is not the best method of collecting useful data. Thus, Sentiment Analysis in social media like Twitter may provide an alternative measure of public opinion and extract useful data (Ceron, Curini, and Stefano 2012; O'Connor et al. 2010; Stieglitz and Dang-Xuan 2012; Zhou et al. 2013). For example, Diakopoulos et al. (2010) present an analysis of ephemeral changes of sentiment in reaction to the first U.S. presidential debate video in 2008.

Further work has been done in other areas of sentiment analysis in social media. Sakaki et al. (2010) propose a method of detecting major events by analysing the stream text in Twitter and at Culotta (2010) propose methods of identifying influenza-related messages. Data from Twitter can be used to analyze public emotion, demography, health characteristics and the "geography of happiness" (Mitchell et al. 2013), a term describing the correlation of sentiment to place. Studying virality in Twitter and the correlation of viral messages with sentiment, Hansen et al. (2011) showed that "news with negative sentiment is more likely to become viral, while in the non-news segment this is not the case".

3. Topic-based Sentiment Analysis Models

We re-implemented a Topic-based Sentiment Analysis (TSA) deep neural network model based on the work of Baziotis et al. (2017). For the purpose of comparing test results, we also re-implemented a Convolutional Neural Network baseline model based on the model of Kim (2014). We named our TSA model Att-BiLSTM+WL (Bidirectional LSTM with Attention mechanism and weighted loss). Before describing the Att-BiLSTM+WL model in depth, we describe the official baselines used in the competition and the strong CNN baseline. Also, we describe a simple bidirectional RNN model, although we have not used it as a baseline, but as part of our model.

3.1 Datasets

Initially, a key prerequisite for creating a good statistical model is having a plethora of data which should also be of good quality. Collecting data and cleaning them is a time consuming process. In our research, training and testing data for the TSA model were provided by the organizers of the SemEval 2017 Task 4 (Rosenthal et al. 2017). The dataset consists of human annotated tweets with sentiment labels in multiple scales with respect to the topic referenced in the message.[7]

As shown in Table 2, the dataset for each subtask consists of:

Subtask B: Tweets, labeled with one out of two class labels, positive or negative sentiment towards that topic.

Subtask C: Tweets, labeled with one out of five class labels, from highly negative, negative, neutral, positive, to highly positive sentiment conveyed by a tweet towards a given topic.

The topics have been selected, based on the ongoing topics (events) trending in Twitter in the period between December 2016 and January 2017 according to TRENDS24[8]**.** The topics cover a range of name entities (e.g., *Fidel Castro, Melania, iPhone, Uber*), geopolitical entities (e.g. *Yemen, Palestine*), and other entities (e.g., *Vaccines, 3D Printing, Dakota Access Pipeline, Thankyouobama, Western media, gun control,* and *vegetarianism*). The developers of the dataset automatically filtered the tweets for duplicates. The construction of the dataset also included a stage that

[7] All annotations for the competition were performed with CrowdFlower (https://www.figure-eight.com/platform)
[8] https://trends24.in

removed the tweets for which the cosine similarity between their Bag of Words representation exceeded the threshold of 0.6 (It is unclear if one of the duplicate tweets was retained in these cases, or if all the duplicates were removed.); also, topics which included less than 100 tweets were discarded. In addition, user profile information was provided such as age and location, as well as friend lists. In our research we didn't study the advantages of including this information (profile information, friend lists) in the data input of the TSA model and we consider it as part of future work.

Sentiment Category		**Positive**		**Neutral**	**Negative**		
5-point label		2	1	0	-1	-2	Total
Train	Subtask B	14897			3997		18894
	Subtask C	1016	12852	12888	3380	296	30432
Test	subtask B	2463			3722		6185
	Subtask C	131	2332	6194	2545	177	12379

Table 2. Dataset statistics for Task 4 subtask B and C.

3.2 Preprocessing

Text preprocessing is a key step towards accurate sentiment classification. The preprocessing of tweets consists of a series of tasks to normalize the text and prepare it for feature extraction. For text preprocessing we will use *ekphrasis*[9], a library for text processing that performs word segmentation, word normalization, tokenization, and spelling correction. The first task is tokenization. In tokenization the text is divided into punctuation, which is often discarded, white space characters, and other tokens that are retained. It's important to keep not only words but also sequences of characters and ciphers that are proven to be useful for sentiment analysis in social media like Twitter (Gimpel et al. 2011). The next task is spelling correction where misspelled words are replaced with the most probable correct word. In addition, word normalization and word segmentation are included in the pipeline. Examples, retrieved from *ekphrasis* documentation, can be found in Table 3. They show the original and the processed text from tweets. During the word normalization most expressions including emoticons, *emojis* or dates, time, currencies, censored words (e.g., f**k, s**t.) along with other special types of tokens are recognized and replaced by labels (e.g., <date>, <phone>, <money>, <user> etc.). These expressions do not usually include information regarding the opinion and don't affect the

[9] https://github.com/cbaziotis/ekphrasis

sentiment; thus, we exclude them from the vocabulary. Normalization also treats hashtags. For instance, #FunFacts becomes <hashtag> fun facts </hashtag>.

Original Tweets
"CANT WAIT for the new season of #TwinPeaks ＼(^o^)／!!! #davidlynch #tvseries :)))"
"I saw the new #johndoe movie and it suuuuucks!!! WAISTED $10... #badmovies :/",
"@SentimentSymp: can't wait for the Nov 9 #Sentiment talks! YAAAAAAY !!! :-D http://sentimentsymposium.com/."
Preprocessed Tweets
cant <allcaps> wait <allcaps> for the new season of <hashtag> twin peaks </hashtag> ＼(^o^)／ ! <repeated> <hashtag> david lynch </hashtag> <hashtag> tv series </hashtag> <happy>
i saw the new <hashtag> john doe </hashtag> movie and it sucks <elongated> ! <repeated> waisted <allcaps> <money> . <repeated> <hashtag> bad movies </hashtag> <annoyed>
<user> : can not wait for the <date> <hashtag> sentiment </hashtag> talks ! yay <allcaps> <elongated> ! <repeated> <laugh> <url>

Table 3. Preprocessing examples.

3.3 Baselines

In this section we describe the official baselines used by the organizers for the task of Topic-based SA. In addition, we report the results by these baselines. We also describe a Convolutional Neural Network (CNN), based on the model of Kim (2014), which we re-implemented to serve as a strong baseline. Finally, we describe a simple bidirectional LSTM model, which is often used as a baseline. Although we did not use this model as a baseline, its description will serve to understand better our TSA ATT-BiLSTM+WL model which will be described in a following section.

The baselines for the Topic-based SA task and for both subtasks are majority-class classifiers. A majority classifier is a classifier that simply labels every instance with the majority class (in the training data) for the corresponding target. As discussed by Rosenthal et al. (2017) “the accuracy of the majority-class classifier is the relative frequency of the majority class, which can be much higher than 0.5 if the test set is imbalanced”. For subtask B the two baselines scores are Baseline

score 1 (B1) for the positive class and B2 for the negative class. Respectively, for subtask C; Baseline score 1 (B1) for the highly negative class, B2 for the negative class, B3 for the neutral class, B4 for the positive class and B5 for the highly positive class. This is how the scores of the baselines as presented in (Rosenthal et al. 2017).

3.3.1 Embedding Layer

A word embedding (Collobert and Weston 2008), (Tang et al. 2014) is a type of word representation that is generated by a model trained on a text corpus, usually with a Language Modeling objective function; additionally, a Text Classification objective may be added. Word representations include semantic and syntactic information of the words. Words are projected in a vector space R^E, where R is a real number and E the dimensions of the embedding layer. The position of a word within the vector space is calculated based on the words that surround it in the text. By retraining word embeddings on a set of data that include sentiment polarity of the words we can use them for sentiment analysis tasks (Maas et al. 2011), (Tang et al. 2014). In a sentiment analysis task, the words are distributed in space according to their sentiment orientation.

At the embedding layer every message is first turned into a vector representation of size equal to the maximum size of all the messages in the dataset (padding to the maximum message length if necessary), where every position $x_1, x_2, \dots, x_T$ is set to an id that points to a word embedding. For instance, the word "*hate*" in the message is turned into "65536" which points to (is the identifier of) a word embedding of size $1 \times R^E$, where R is a real number and E the dimensions of the embedding layer.

3.3.2 Convolutional neural network baseline

We built a Convolutional Neural Network (LeCun et al. 1989) model based on the work of Kim on sentence classification (Kim 2014). Here we describe the CNN model we re-implemented as a strong baseline. In the Convolutional Neural Network (CNN) model, first, we pass the words of the message $X^{tw} = \left(x_1^{tw}, x_2^{tw}, \dots, x_{T_{tw}}^{tw}\right)$ and the words of the topic $X^{to} = \left(x_1^{to}, x_2^{to}, \dots, x_{T_{to}}^{to}\right)$ through an embedding layer, as in Section 3.3.1. On the embedding layer, we added noise and we applied dropout (Gal and Ghahramani 2016) to prevent overfitting (in Section 3.4.3 we provide more information on noise and dropout). Second, we concatenated the

word vectors of the message with the word vectors of the topic to create a single sentence matrix representation $A \in R^{sE}$, where s the length of the concatenated sentences and E the dimensions of the embedding layer, and we used A_{ij} or $A[i{:}j]$ to indicate the sub-matrix of A from row i to row j. We applied a convolution operation using multiple filters $w \in R^{hE}$, where h is the size of the window of the filter and E the dimensions of the embedding layer. The filters we use have different window sizes h in effect considering n-grams of different lengths. The output $c_i \in R^{s-h+1}$ of the convolution is produced by repeatedly applying the filters on the sub-matrices of A, where $1 \leq \mathrm{i} \leq s-h+1$, f is an activation function, $\cdot$ is the dot product between the sub-matrix and the filters and $b \in R$ is a bias term. Each filter is applied across the sentence matrix A_{ij} successively and produces a feature map.

$$c_i = f(w \cdot A[i : i+h-1] + b)$$

$$c = [c_1, c_2, \ldots c_{n-h+1},]$$

Next, we applied max over time pooling (Collobert et al. 2011) over the different feature maps to get the maximum value $\tilde{c} = max\{c_1, c_2, \ldots c_{n-h+1}\}$ from each one. In each map the feature with the highest value corresponds to the most important features. After applying dropout, the max features obtained from all filters are passed through a fully connected softmax layer, as shown in Figure 1, that produces a probability distribution over all possible target classes. For optimization we used Adam optimizer (Kingma and Ba 2014). In Section 3.4.4 we provide more information on the optimization method that we used in both models. For the number of kernels (filters) and the kernel sizes, we used the corresponding hyperparameter values of Kim (2014). For the dropout rate and noise parameters we used the following values.

CNN model hyperparameters for both subtasks:

- Embedding Layer size: 200.
- Kerner Dimensions: 100
- Kernel sizes: $\{3, 4, 5\}$
- Gaussian Noise for Embedding Layer: $\sigma = 0.5$.
- Embedding Layer dropout factor: 0.5.

- Optimizer learning rate: 0.001.
- Gradient Norm clipping: 1.
- Mini-Batches of size: 128.

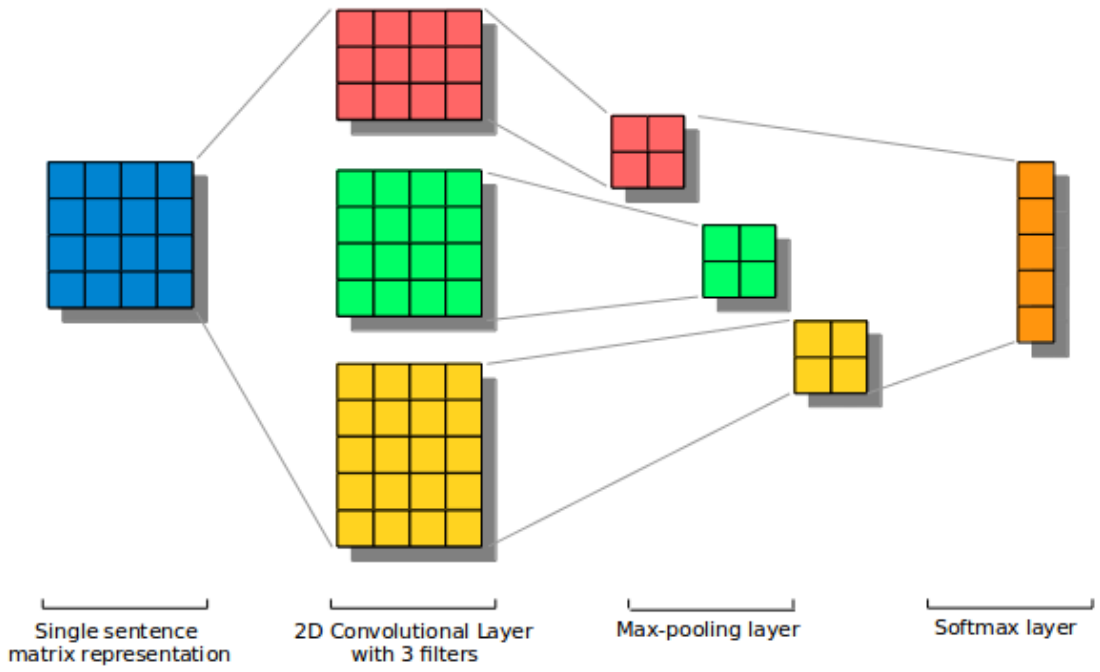

Figure 1. CNN model with 3 filters.

3.3.3 Bidirectional LSTM baseline

A Long Sort Term Memory (LSTM) is a variation of recurrent neural networks, capable of learning long-term dependencies, that was proposed by Hochreiter and Schmidhuder (1997) as a solution to vanishing gradients. An LSTM allows the recurrent nets to continue to learn over time, by preserving the error through time during backpropagation, and thus creating a memory mechanism. A standard LSTM network is composed of many memory blocks known as cells. Each cell can be trained to decide which information from the sequence should be saved, propagated to the output or be discarded. The LSTM block consists of:

- The forget gate f_t, responsible for discarding information from the cell state that is not required or has no importance in understanding data.

- An input gate i_t, for adding information to the cell state.
- The output gate o_t, which is responsible for selecting the information that is useful to be shown in the current cell state.

The LSTM block is described by the following set of equations, where σ stands for a sigmoid activation:

Gates:

$$f_t = \sigma\left(W_f x_t + U_f h_{t-1} + b_f\right)$$

$$i_t = \sigma\left(W_i x_t + U_i h_{t-1} + b_i\right)$$

$$o_t = \sigma\left(W_o x_t + U_o h_{t-1} + b_o\right),$$

Input transform:

$$c_t = f_t \circ c_{t-1} + i_t \circ tanh(W_c x_t + U_c h_{t-1} + b_c)$$

State update:

$$h_t = o_t \cdot tanh(c_t)$$

In order to get more information from the sequence of the text that we use as an input to the LSTM, we make two passes over the sequence, one from the left to the right or from x_i to x_T and one from the right to the left or from x_T to x_i. We concatenate the forward LSTM $\vec{f}$ with the backward LSTM $\overleftarrow{f}$ into a representation:

$$h_i = \overrightarrow{h_i} \parallel \overleftarrow{h_i}, \qquad h_i \in R^L$$

where L is the dimension of the LSTM.

3.4 Att-BiLSTM+WL model

For topic-based sentiment analysis we re-implemented Baziotis et al. (2017) model, a deep learning model using a bidirectional LSTM recurrent neural network with attention mechanism and weighted loss (Att-BiLSTM+WL). In contrast with Baziotis et al. model, we did not used a Maxout layer. Given a message and the corresponding topic for the message the Att-BiLSTM+WL model can classify the tweet in a binary sentiment polarity (i.e., positive - negative) and in a five-scale sentiment polarity towards the topic (i.e., highly positive, positive, neutral, negative, highly

negative). The model architecture includes an embedding layer, a pair of two-level deep Bi-LSTMs with shared weights and an attention layer. We used word embeddings of 200 dimensions from Glove (Pennington, Socher, and Manning 2014) pre-trained on Twitter. The output of the embedding layer, which is of size (max $size \times 200$), is used as an input for the bidirectional LSTM.

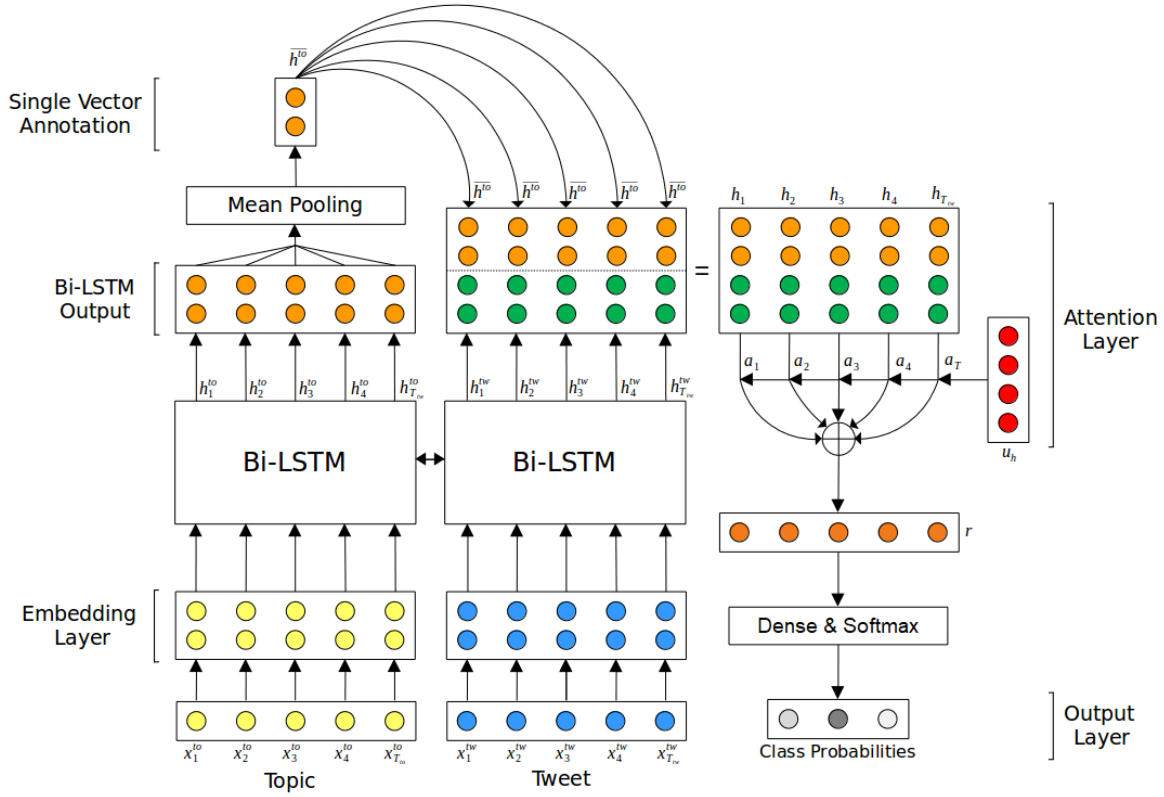

Figure 2. The Att-BiLSTM+WL model: A 2-layer bidirectional LSTM with attention.

Given that the task in hand provides us with the message and the topic regarding the message, we use two inputs for the Bi-LSTM as shown in Figure 2. After passing the words of the message $X^{tw} = (x_1^{tw}, x_2^{tw}, \dots, x_{T_{tw}}^{tw})$ and the words of the topic $X^{to} = (x_1^{to}, x_2^{to}, \dots, x_{T_{to}}^{to})$ through the embedding layer we use those as two inputs to the Bi-LSTM with shared weights. The Bi-LSTM, then, produces the hidden representation of the message $H^{tw} = (h_1^{tw}, h_2^{tw}, \dots, h_{T_{tw}}^{tw})$ and of the topic $H^{to} = (h_1^{to}, h_2^{to}, \dots, h_{T_{to}}^{to})$.

In the next sections, we describe the layers of the Att-BiLSTM+WL model that follow on from the output of the bidirectional LSTM and the important procedures of model regularization and optimization. We also report the hyperparameters of the Att-BiLSTM+WL model.

3.4.1 Mean Pooling and Annotation

At the Mean Pooling layer, in order to produce the vector of the topic $\overline{h^{to}}$ we compute the mean over time of all hidden states of $H^{to} = (h_1^{to}, h_2^{to}, \dots, h_{T_{to}}^{to})$.

$$\overline{h^{to}} = \frac{1}{T_{to}} \sum_{1}^{T_{to}} h_i^{to}$$

We use this single vector annotation $\overline{h^{to}}$ and concatenate it with each one of the hidden states of vector H^{tw}.

$$h_i = h_i^{tw} \parallel \overline{h^{to}}, \qquad h_i \in R^{2L}$$

where L is the dimension of the LSTM. This way we create vectors that incorporate the information of the topic and each word in the message.

3.4.2 Context Attention Layer

In any text there are some words that carry more emotional weight than others. These words are the ones that shift the polarity of the message. In order to find the most important words we use an attention mechanism as in Seo et al. (2016). This is the Attention Layer as shown in Figure 2. The mechanism calculates for each word a weight, which we use to weigh the participation of the words in the final representation of the message. The word representation h_i is passed through a dense layer with $tanh$ activation, and the resulting vector is multiplied with a context vector u_h (learned during backpropagation) to obtain the attention score a_i of h_i. A softmax is applied to the attention scores to make them sum to 1. The final representation r of the message-topic concatenation is the weighted (by the attention scores) sum of all the world annotations h_i. Specifically:

$$e_i = \tanh(W_h h_i + b_h), \qquad e_i \in [-1,1]$$

$$a_i = \frac{\exp(e_i\, u_h)}{\sum_{t=1}^{T} \exp(e_i\, u_h)}, \qquad \sum_{i=1}^{T} a_i = 1$$

$$r = \sum_{i=1}^{T} a_i h_i \ , \qquad r \in R^{2L}$$

where W_h , b_h and u_h are learned through training.

3.4.3 Regularization

Before we start training we apply regularization to the Att-BiLSTM+WL model. Regularization is key to control and fine-tune the model's complexity. If the model is very complex it will overfit the training data which will lead to misclassifying the test data. On the other hand, if the model is too simple then it will have low training accuracy. This is called underfitting. For this purpose, we apply two regularization methods. We apply Gaussian noise (Rasmussen 2004) to the embedding layer, as a dataset augmentation method. Also, we apply dropout to the embedding layer to prevent overfitting (Geoffrey E Hinton et al. 2012), which is a technique that prevents units from co-adapting excessively to the data, by dropping units from the neural network randomly during training. By applying dropout both in the embedding layer and the RNN as in (Gal and Ghahramani 2016) we expect to improve the performance of our Att-BiLSTM+WL model that learns more robust features.

3.4.4 Optimization

We trained our neural network model on the 90% of the training dataset and we kept the 10% for validation. All the models we compare we used the same training and validation data. We used the Cross-Entropy Loss function (Golik, Doetsch, and Ney 2013). To optimize the model's performance we used the Adam optimizer (Kingma and Ba 2014), which is an extension of Stochastic Gradient Descent. The Stochastic Gradient Descent (SGD) is an optimization algorithm used to update the weights of the network in order to minimize the cost.

Cross-Entropy Loss:

$$E(y, \acute{y}) = \sum_{i=1}^{K} \acute{y}_i \log(y_i)$$

where y_i is the predicted probability distribution of the class i and $\acute{y}_i$ is the true label of the class, which is 0 or 1.

During training, the weights of the network are updated using the following equation:

$$W = W - \alpha \nabla J(W, b)$$

Stochastic Gradient Descent (with mini-batches of size m):

$$J(W,b) = \frac{1}{m}\sum_{i=0}^{m} J(W,b,x^{(i)},y^{(i)})$$

where W are the weights, α is the learning rate, ∇ is the gradient of the cost function $J(W,b)$ with respect to changes in the weights and m is the number of training instances ($x^{(i)}$ and $y^{(i)}$) in a mini-batch. In our task we exploited *Mini-batch Gradient Descent*, where the gradient is averaged over a small number of samples from the dataset which are called batches. We trained the model using mini-batches of size 128.

In addition to the classic methods for updating the network's weights, there are also more sophisticated alternatives that have been proposed in the last years such as AdaGrad (Duchi, Bartlett, and Wainwright 2012), AdaDelta (Zeiler 2012) and Adam (Kingma and Ba 2014) , which was employed in this work. These methods integrate the automated adjustment of the learning rates during the training phase. Adam is an optimization algorithm introduced as a variation of the SGD which, as described by the authors, can handle sparse gradients and noisy data. In detail, Adam offers a combination of the AdaGrad and the RMSProp (Geoffrey E. Hinton, Srivastava, and Swersky 2012) optimizer.

Also, during training we performed gradient clipping to minimize the phenomenon of exploding gradients (Pascanu, Mikolov, and Bengio 2012).

3.4.5 Hyper-parameters

In the interest of training a model that can achieve the best scores, suitable hyper-parameter tuning is required. For the Att-BiLSTM+WL model we manually tuned the hyperparameters based on previous work (Baziotis et al. 2017), that uses a combination of Random Search (Strub et al. 2017) and Bayesian Optimization (Snoek, Larochelle, and Adams 2016). For the dropout rate and noise parameters, we adopted the defaults of the implementations we used.

Att-BiLSTM+WL model hyperparameters for both subtasks:

- Embedding Layer size: 200.
- RNN Hidden Layer size: 150.
- Gaussian Noise for Embedding Layer: $\sigma = 0.5$.

- Embedding Layer Dropout factor: 0.5.
- RNN dropout factor: 0.5.
- Optimizer learning rate: 0.001.
- Gradient Norm clipping: 1.
- Mini-Batches of size: 128.

3.5 Class Balancing

A very common problem in machine learning classification tasks is the case where there are distinct classes with higher number of instances compared to other classes in the dataset. This phenomenon is called class imbalance. For instance, in subtask C the model is more likely to classify a message as neutral due to the number of instances for the particular class. In order to cope with this problem, we add weights in the loss function to punish the misclassification of the classes with the fewer instances. This way the loss is higher for under-represented classes and lower for the over-represented ones. To compute the weights of each class we use the following equation[10]:

$$w_i = \frac{n}{kn_i}$$

where w_i is the weight of class i, n is the number of all observations, n_i is the number of observations in class i and k is the total number of classes.

The weighted loss function is described by the following formula:

$$E(y, \acute{y}_i) = \sum_{i=1}^{k} w_i \, \acute{y}_i \, log \, (y_i)$$

where w_i is the weight assigned to each class where y_i is the predicted probability distribution of the class i and $\acute{y}_i$ is the true label of the class, which is 0 or 1. In Section 4.4 you can see the results of the class balancing with the introduction of the weighted loss function.

[10] http://scikit-learn.org/stable/modules/generated/sklearn.utils.class_weight.compute_class_weight.html

4. Experiments

4.1 Evaluation Measures

Here we present the evaluation measures for Sentiment Analysis in Twitter as given in SemEval 2017 Task 4 (Rosenthal et al. 2017).

For **Subtask B** the evaluation measures are accuracy or *Acc*, macro-averaged Recall and macro-averaged $F1$.

		Predicted Class	
		P	N
Actual Class	P	True Positives (TP)	False Negatives (FN)
	N	False Positives (FN)	True Negatives (TN)

Table 4. Confusion Matrix

Accuracy is the ratio of correctly predicted instances, where an instance is a text message in Twitter. In Table 4 we present the confusion matrix for the positive class for a binary classifier, as the one used in subtask B; a confusion matrix for the negative class also applies.

In more detail:

True Positives (TP): Predicted as positive instances that were actually positive.

True Negatives (TN): Predicted as negative instances that were actually negatives.

False Positives (FP): Predicted as positive but were negative instances.

False Negatives (FN): Predicted as negative but were positive instances.

$$Acc = \frac{TP + TN}{TP + TN + FP + FN}$$

R_{macro} score is the macro-averaged score of Recall in all classes (positive, negative).

$$R^P = \frac{TP}{TP + FN}$$

$$R^N = \frac{TN}{TN + FP}$$

$$R_{macro} = \frac{1}{2}\,(R^P + R^N)$$

$F1_{macro}$ is the macro-averaged score (over the positive and negative class) $F1$. To calculate $F1^P$ for positives, we need to calculate the corresponding precision P^P, where P^P is the ratio of messages predicted positive that were actually positive.

$$P^P = \frac{TP}{TP + FP}$$

$$F1^P = \frac{2 \times P^P \times R^P}{P^P \times R^P}$$

likewise, we calculate $F1^N$

$$F1^N = \frac{2 \times P^N \times R^N}{P^N \times R^N}$$

as follows:

$$F1_{macro} = \frac{1}{2}(F1^P + F1^N)$$

For **Subtask C** the predictions are on a five-point scale, $C = \{-2, -1, 0, +1, +2\}$, where -2 is for highly negative, -1 for negative, 0 for negative, $+1$ for positive and $+2$ for highly positive. For example, classifying a message as positive $(+1)$ produces bigger error if the actual class is highly negative (-2) than if it is negative (-1) . Therefore, in Subtask C the evaluation measures are macro-averaged mean absolute error (MAE^M) and micro-averaged mean absolute error MAE^μ. Lower values are better.

$$MAE^M(h, T_e) = \frac{1}{|C|} \sum_{j=1}^{|C|} \frac{1}{\left|T_{e_j}\right|} \sum_{x_i \in T_{e_j}} |\, h(x_i) - y_i \,|$$

$$MAE^{\mu}(h, T_e) = \frac{1}{|T_e|} \sum_{x_i \in T_e} | \, h(x_i) - y_i \, |$$

where y_i is the actual class of x_i, $h(x_i)$ is the predicted class of x_i and $| \, h(x_i) - y_i \, |$ is the absolute error between predicted and actual class. T_{e_j} is the set of all observations in test set T_e that are of class c_j.

MAE^{μ} is the mean absolute error for all for all the observations in the test set, while MAE^{M} is the mean error of all classes. MAE^{M} is a better measure for subtask C than MAE^{μ} since it takes into account the class imbalance of the observations.

4.2 Experimental Setup

To implement the models, we used PyTorch[11] with Theano (Al-Rfou et al. 2016) as backend. We trained the neural networks on a GTX 1060 (6GB) GPU. We provide the source code of the Att-BiLSTM+WL and CNN model through GitHub[12]

4.3 Evaluation at Validation

Before we present the scores that the Att-BiLSTM+WL model achieved in the task, we explain how we worked during training and the decisions we made based on evaluations over the validation data. In Subtask B we trained the model for 30 epochs, while evaluating its progress by measuring the accuracy on the validation part of the dataset for the same number of epochs. We decided to use accuracy as our main evaluation metric hence we stopped the training after 17 epochs (as shown in Figure 3, the point where the model achieved optimum accuracy). We evaluated and compared the Att-BiLSTM+WL model with the CNN model for 30 epochs and we observed (Figure 4) that the CNN achieves its best performance after the 25th epoch, but, still achieves worse accuracy than the Att-BiLSTM+WL at its best epoch (17th), though the difference is small and the performance of the CNN is more stable. There is no need to train beyond 30 epochs because the validation metrics decline for both models from that point on. Likewise, for Subtask C we trained the model for 18 epochs considering that this was the lowest point for

[11] https://pytorch.org/
[12] https://github.com/kkorovesis/Att-BiLSTM-WL

MAE^M during validation (Figure 5). We chose to fine-tune the Att-BiLSTM+WL model with respect only to macro-averaged MAE, but in future work we would like to study the case of a more balanced performance for both micro and macro averaged MAE. Here the Att-BiLSTM+WL performs significantly better than the CNN model (Figure 6).

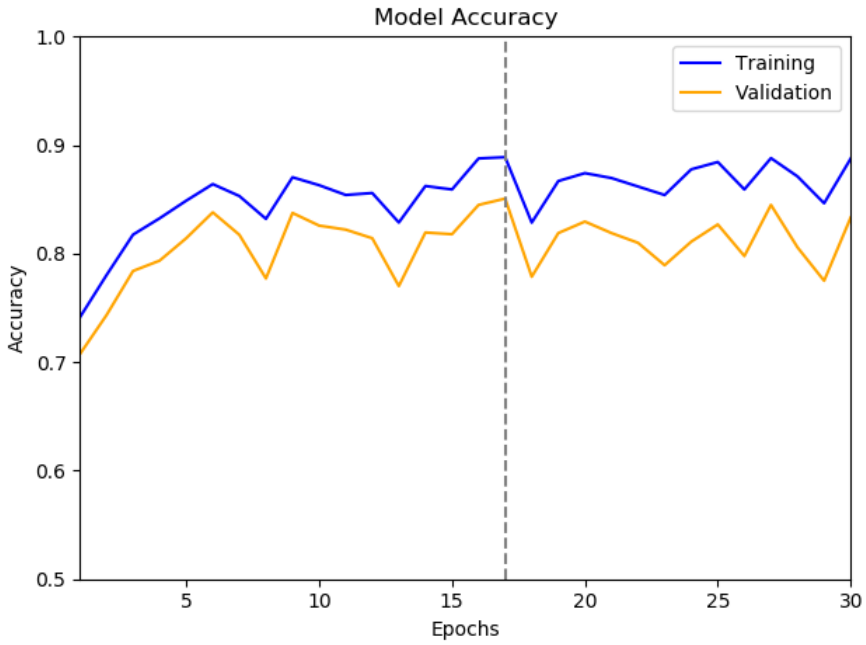

Figure 3. Att-BiLSTM+WL Accuracy in 30 epochs, subtask B.

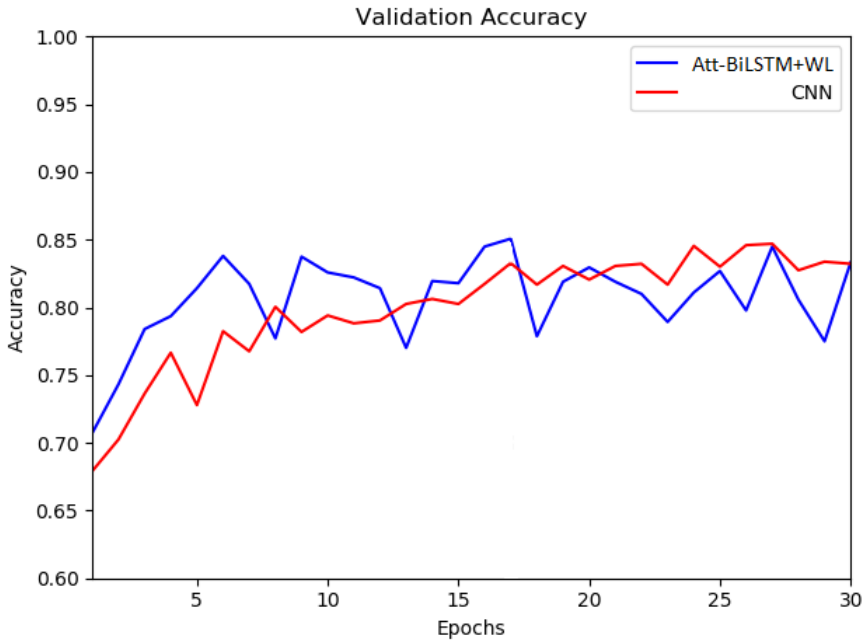

Figure 4. Validation Accuracy of Att-BiLSTM+WL and CNN model, subtask B.

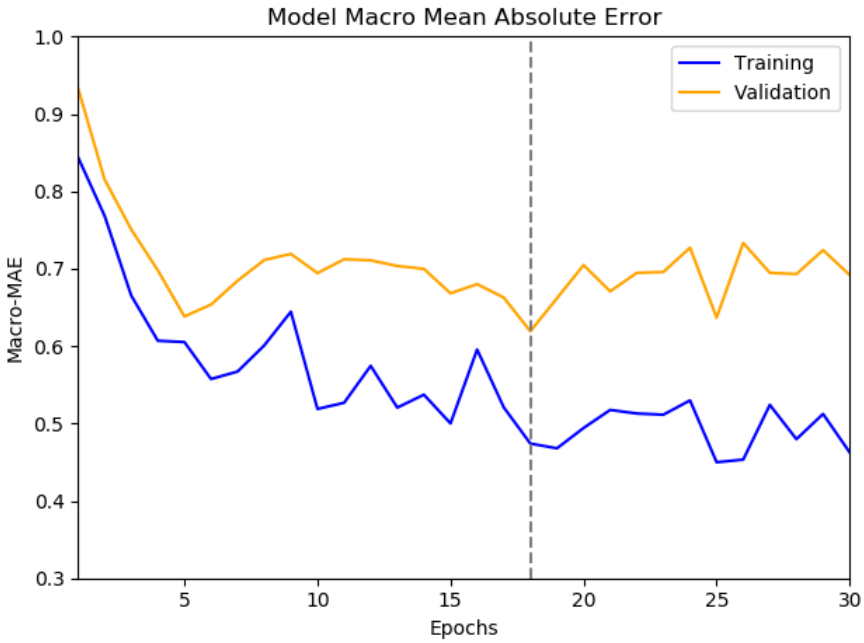

Figure 5. Att-BiLSTM+WL macro-MAE in epochs, subtask C (lower is better).

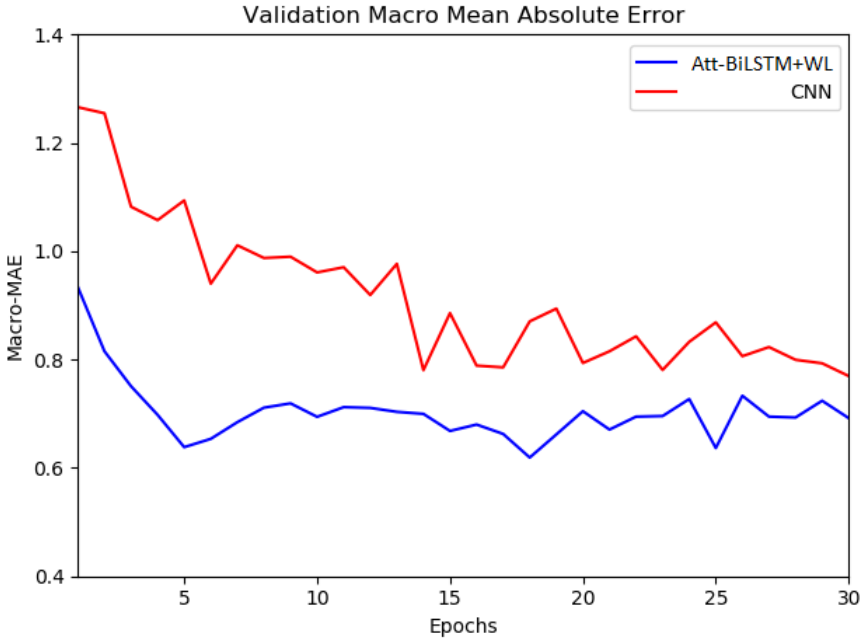

Figure 6. Validation macro-MAE of Att-BiLSTM+WL and CNN model, subtask C (lower is better).

4.4 Evaluation at Test

For Subtask B, the Att-BiLSTM+WL model outperforms the CNN model in Accuracy and $F1_{macro}$ on the test set (Table 5). We observed that both models score very high compared to the SemEval 2017 baselines (Rosenthal et al. 2017). In addition, for Subtask C the Att-BiLSTM+WL model achieved very high scores on test (see Table 6). Based on MAE^M (the main ordinal classification measure, as described in Section 4.1), the Att-BiLSTM+WL outperforms the CNN model and all the baselines. We employed weighted loss function (see Section 3.5) on the BiLSTM model to study the effects for both subtasks. In Subtask B, we improved the accuracy score and $F1_{macro}$ score but the score for R_{macro} was significantly decreased (see Table 7, where Att-BiLSTM represents the model without the weighted loss function). Here the model has misclassified more true positives (TP) and thus the R_{macro} is lower. The model is more accurate in the class with the smaller coverage but fails in the class with the bigger coverage due to the weighted loss function. In Subtask C, by employing WL in the model we improved the MAE^M score and R_{macro} score. This was expected due to the considerable class imbalance of the dataset (see Table 8). On the other hand, the Att-BiLSTM with WL scores lower in accuracy and MAE^μ because these metrics don't consider the class ratio. We observe that despite the weighted loss function the $F1_{macro}$ is lower. Based on former observations, we expected all the macro averaged scores to be higher when employing WL. We assume that in this subtask the model has lower precision in specific classes, because of the weights w_i of the classes, and that effects the $F1_{macro}$. To prove this assumption, more experiments and error analysis is required. We consider this part for our future work.

System	Acc	$F1_{macro}$	R_{macro}
Att-BiLSTM+WL	**0.861**	**0.854**	0.808
CNN	0.771	0.769	**0.862**
B1-All POSITIVE	0.398	0.285	0.500
B2-All NEGATIVE	0.602	0.376	0.500

Table 5. Scores in Subtask B (higher is better), Bx indicates a baseline.

System	MAE^M	MAE^μ
Att-BiLSTM+WL	**0.585**	**0.898**
CNN	0.829	1.167
B1-HIGHLY NEGATIVE	2.000	1.895
B2-NEGATIVE	1.400	0.923
B3-NEUTRAL	1.200	0.525
B4-POSITIVE	1.400	1.127
B5-HIGLY POSITIVE	2.000	2.105

Table 6. Scores in Subtask C (lower is better), Bx indicates a baseline.

System	Acc	$F1_{macro}$	R_{macro}
Att-BiLSTM+WL	**0.861**	**0.854**	0.808
Att-BiLSTM	0.843	0.841	**0.892**
CNN	0.771	0.769	0.862

Table 7. The effect of weighted loss (WL) in Subtask B (higher is better).

System	MAE^M	MAE^μ	Acc	$F1_{macro}$	R_{macro}
Att-BiLSTM+WL	**0.585**	0.898	0.423	0.319	**0.555**
Att-BiLSTM	0.608	**0.813**	**0.494**	**0.373**	0.508
CNN	0.829	1,167	0.236	0.201	0.457

Table 8. The effect of weighted loss (WL) in Subtask C (for MAE lower is better).

We note here that these results can improve even further by employing better fine tuning. [13]

[13] The fine-tuned model, described in (Baziotis et al. 2017) achieved slightly better results; i.e., 0.897$_1$ Acc, 0.890$_1$ $F1_{macro}$ and 0.882$_1$ R_{macro} in subtask B; 0.555$_2$ MAE^M and 0.543$_4$ MAE^μ in subtask C.

5. Conclusions and Future Work

5.1 Conclusions

In this thesis we built a neural network model (called Att-BiLSTM+WL) for topic-based sentiment classification for messages in Twitter. We re-implemented in PyTorch a Bi-LSTM model (Att-BiLSTM+WL) based on the work of Baziotis et al. (2017) for a sentiment analysis task that consists of two subtasks and we released it for public use. We managed to outperform the baselines provided by SemEval 2017, while scoring high results in both subtasks. We obtained a test accuracy score of 0.860 in subtask B and regarding subtask C we reduced the macro-average mean absolute error in test data at 0.584. In addition, we built and trained a CNN model (Kim 2014) and compared results obtained from both models. The Att-BiLSTM+WL performs slightly better than the CNN model in subtask B and much better in subtask C, see (Table 5 and Table 6). We added a weighted loss to Att-BiLSTM, leading to the Att-BiLSTM+WL and studied the effect in both subtasks. Although this is not a novel addition, we evaluated this component reporting the margin with which it improves the model.

5.2 Future work

More research will help deliver even better results in similar tasks, while other types of neural network, such as CNNs are also getting very good results alone or by working together with LSTMs (Cliche 2017). We did not extensively tune the hyper-parameters of our models. In most cases, we used defaults or hyper-parameter values from previous work. Hence, further improvements may be possible with hyper-parameter tuning, for example using Bayesian Optimization (Snoek et al. 2016). In the Att-BiLSTM+WL model we “froze” the embeddings, not letting their weights to be updated during training. As a next step, we intend to study employing trainable word embeddings, in order to examine whether better and domain adapted word representations can improve the models. Therefore, future work consists of:

- Extensive fine tuning.
- Trainable embeddings.

Bibliography

Al-Rfou, Rami et al. 2016. “Theano: A Python Framework for Fast Computation of Mathematical Expressions.” *CoRR* abs/1605.0.

Appel, Orestes, Francisco Chiclana, Jenny Carter, and Hamido Fujita. 2016. “A Hybrid Approach to the Sentiment Analysis Problem at the Sentence Level.” *Knowledge-Based Systems* 108:110–24.

Arias, Marta, Argimiro Arratia, and Ramon Xuriguera. 2013. “Forecasting with Twitter Data.” *ACM Transactions on Intelligent Systems and Technology* 5(1):1–24.

Asur, Sitaram and Bernardo A. Huberman. 2010. “Predicting the Future with Social Media.” Pp. 492–99 in *2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, *WI-IAT ’10*. Washington, DC, USA: IEEE Computer Society.

Baziotis, Christos, Nikos Pelekis, and Christos Doulkeridis. 2017. “DataStories at SemEval-2017 Task 4: Deep LSTM with Attention for Message-Level and Topic-Based Sentiment Analysis.” *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)* (1):747–54.

Bollen, Johan, Huina Mao, and Xiaojun Zeng. 2011. “Twitter Mood Predicts the Stock Market.” *Journal of Computational Science* 2(1):1–8.

Ceron, Andrea, Luigi Curini, and M. Stefano. 2012. “Tweet Your Vote: How Content Analysis of Social Networks Can Improve Our Knowledge of Citizens’ Policy Preferences. An Application to Italy and France.” *New Media &amp Society* 16:1–24.

Chevalier, Judith A. and Dina Mayzlin. 2006. “The Effect of Word of Mouth on Sales: Online Book Reviews.” *Journal of Marketing Research* 43(3):345–54.

Collobert, Ronan et al. 2011. “Natural Language Processing (Almost) from Scratch.” *CoRR* abs/1103.0398.

Collobert, Ronan and Jason Weston. 2008. “A Unified Architecture for Natural Language Processing.” Pp. 160–67 in *Proceedings of the 25th international conference on Machine learning - ICML ’08*, *ICML ’08*. New York, NY, USA: ACM.

Culotta, Aron. 2010. “Detecting Influenza Outbreaks by Analyzing Twitter Messages.” Pp. 115–

22 in *Proceedings of the First Workshop on Social Media Analytics*, *SOMA '10*. New York, NY, USA: ACM.

Diakopoulos, Nicholas A. and David A. Shamma. 2010. "Characterizing Debate Performance via Aggregated Twitter Sentiment." P. 1195 in *Proceedings of the 28th international conference on Human factors in computing systems - CHI '10*, *CHI '10*. New York, NY, USA: ACM.

Duchi, John C., Peter L. Bartlett, and Martin J. Wainwright. 2012. "Randomized Smoothing for (Parallel) Stochastic Optimization." *Proceedings of the IEEE Conference on Decision and Control* 12:5442–44.

Gal, Yarin and Zoubin Ghahramani. 2016. "A Theoretically Grounded Application of Dropout in Recurrent Neural Networks." Pp. 1027–35 in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, *NIPS'16*. USA: Curran Associates Inc.

Gilbert, Eric and Karrie Karahalios. 2010. "Widespread Worry and the Stock Market." *Proceedings of the 4th International AAAI Conference on Weblogs and Social Media* 58–65.

Gimpel, Kevin et al. 2011. "Part-of-Speech Tagging for Twitter: Annotation, Features, and Experiments." Pp. 42–47 in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers - Volume 2*, *HLT '11*. Stroudsburg, PA, USA: Association for Computational Linguistics.

Goldberg, Yoav and Graeme Hirst. 2017. *Neural Network Methods in Natural Language Processing*. Morgan & Claypool Publishers.

Golik, Pavel, Patrick Doetsch, and Hermann Ney. 2013. "Cross-Entropy vs. Squared Error Training: A Theoretical and Experimental Comparison." *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH* 1756–60.

Han, Bing. 2008. "Investor Sentiment and Option Prices." *Review of Financial Studies* 21(1):387–414.

Hansen, Lars Kai, Adam Arvidsson, Finn Aarup Nielsen, Elanor Colleoni, and Michael Etter. 2011. "Good Friends, Bad News - Affect and Virality in Twitter." *Communications in Computer and Information Science* 185 CCIS(PART 2):34–43.

Hinton, Geoffrey E., Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan

Salakhutdinov. 2012. “Improving Neural Networks by Preventing Co-Adaptation of Feature Detectors.” *CoRR* abs/1207.0580.

Hinton, Geoffrey E., Nitish Srivastava, and Kevin Swersky. 2012. “Lecture 6e- Rmsprop: Divide the Gradient by a Running Average of Its Recent Magnitude.” *COURSERA: Neural Networks for Machine Learning* 26–31.

Ho, Tin Kam. 1995. “Random Decision Forests.” Pp. 278–82 vol.1 in *Proceedings of 3rd International Conference on Document Analysis and Recognition*. Vol. 1.

Hochreiter, Sepp and Jürgen Schmidhuber. 1997. “Long Short-Term Memory.” *Neural Computation* 9(8):1735–80.

Hu, Minqing and Bing Liu. 2004a. “Mining and Summarizing Customer Reviews.” *Proceedings of the 2004 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD ’04* 168.

Hu, Minqing and Bing Liu. 2004b. “Mining and Summarizing Customer Reviews.” P. 168 in *Proceedings of the 2004 ACM SIGKDD international conference on Knowledge discovery and data mining - KDD ’04*, *KDD ’04*. New York, NY, USA: ACM.

Jansen, Bernard J., Mimi Zhang, Kate Sobel, and Abdur Chowdury. 2009. “Twitter Power: Tweets as Electronic Word of Mouth.” *Journal of the American Society for Information Science and Technology* 60(11):2169–88.

Kim, Yoon. 2014. “Convolutional Neural Networks for Sentence Classification.” *CoRR* abs/1408.5882.

Kingma, Diederik P. and Jimmy Ba. 2014. “Adam: A Method for Stochastic Optimization.” *CoRR* abs/1412.6.

LeCun, Y. et al. 1989. “Backpropagation Applied to Handwritten Zip Code Recognition.” *Neural Computation* 1(4):541–51.

Lemmon, Michael and Evgenia Portniaguina. 2006. “Consumer Confidence and Asset Prices: Some Empirical Evidence.” *Review of Financial Studies* 19(4):1499–1529.

Liu, Yang, Xiangji Huang, Aijun An, and Xiaohui Yu. 2007. “Arsa.” P. 607 in *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR ’07*, *SIGIR ’07*. New York, NY, USA: ACM.

Maas, Andrew L. et al. 2011. “Learning Word Vectors for Sentiment Analysis.” *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies* 142–50.

Makrehchi, Masoud, Sameena Shah, and Wenhui Liao. 2013. “Stock Prediction Using Event-Based Sentiment Analysis.” Pp. 337–42 in *Proceedings - 2013 IEEE/WIC/ACM International Conference on Web Intelligence, WI 2013*. Vol. 1, *WI-IAT ’13*. Washington, DC, USA: IEEE Computer Society.

Martinez-Arroyo, M. and L. E. Sucar. 2006. “Learning an Optimal Naive Bayes Classifier.” P. 958 in *18th International Conference on Pattern Recognition (ICPR’06)*. Vol. 4.

Mishne, Gilad and Natalie Glance. 2006. “Predicting Movie Sales from Blogger Sentiment.” *AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs* 155–58.

Mitchell, Lewis, Morgan R. Frank, Kameron Decker Harris, Peter Sheridan Dodds, and Christopher M. Danforth. 2013. “The Geography of Happiness: Connecting Twitter Sentiment and Expression, Demographics, and Objective Characteristics of Place.” *PLoS ONE* 8(5):1–15.

O’Connor, Brendan, Ramnath Balasubramanyan, Bryan R. Routledge, and Noah A. Smith. 2010. “From Tweets to Polls: Linking Text Sentiment to Public Opinion Time Series.” *International AAAI Conference on Weblogs and Social Media* 11.

Oh, Chong and Olivia Sheng. 2011. “Investigating Predictive Power of Stock Micro Blog Sentiment in Forecasting Future Stock Price Directional Movement.” in *ICIS*.

Pang, Bo and Lillian Lee. 2005. “Seeing Stars: Exploiting Class Relationships for Sentiment Categorization with Respect to Rating Scales.” *CoRR* abs/cs/0506075.

Pang, Bo, Lillian Lee, and Shivakumar Vaithyanathan. 2002. “Thumbs Up?” *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - EMNLP ’02* 10:79–86.

Pascanu, Razvan, Tomas Mikolov, and Yoshua Bengio. 2012. “On the Difficulty of Training Recurrent Neural Networks.” P. III-1310--III-1318 in *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*, *ICML’13*. JMLR.org.

Pennington, Jeffrey, Richard Socher, and Christoper Manning. 2014. “Glove: Global Vectors for Word Representation.” *EMNLP* 14:1532–43.

Popescu, Ana-Maria and Oren Etzioni. 2005. “Extracting Product Features and Opinion from Reviews.” Pp. 339–46 in *Human Language Technology and Empirical Methods in Natural Language Processing, Vancouver, British Columbia*, *HLT ’05*. Stroudsburg, PA, USA: Association for Computational Linguistics.

Rasmussen, Carl Edward. 2004. “Gaussian Processes in Machine Learning.” Pp. 63–71 in *Advanced Lectures on Machine Learning: ML Summer Schools 2003, Canberra, Australia, February 2 - 14, 2003, Tübingen, Germany, August 4 - 16, 2003, Revised Lectures*, edited by O. Bousquet, U. von Luxburg, and G. Rätsch. Berlin, Heidelberg: Springer Berlin Heidelberg.

Rosenthal, Sara, Noura Farra, and Preslav Nakov. 2017. “SemEval-2017 Task 4: Sentiment Analysis in Twitter.” *Proceedings of the 11th International Workshop on Semantic Evaluations (SemEval-2017)* 502–18.

Sakaki, Takeshi, Makoto Okazaki, and Yutaka Matsuo. 2010. “Earthquake Shakes Twitter Users.” P. 851 in *Proceedings of the 19th international conference on World wide web - WWW ’10*, *WWW ’10*. New York, NY, USA: ACM.

Seo, Paul Hongsuck, Zhe Lin, Scott Cohen, Xiaohui Shen, and Bohyung Han. 2016. “Progressive Attention Networks for Visual Attribute Prediction.” *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* 1480–89.

Si, Jianfeng et al. 2013. “Exploiting Topic Based Twitter Sentiment for Stock Prediction.” *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* (2011):24–29.

Smailović, Jasmina, Miha Grčar, Nada Lavrač, and Martin Žnidaršič. 2013. “Predictive Sentiment Analysis of Tweets: A Stock Market Application.” *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 7947 LNCS:77–88.

Snoek, Jasper, Hugo Larochelle, and Rp Adams. 2016. “Practical Bayesian Optimization of Machine Learning Algorithms.” Pp. 2951–59 in *Nips*, edited by F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger. Curran Associates, Inc.

Sprenger, Timm O., Andranik Tumasjan, Philipp G. Sandner, and Isabell M. Welpe. 2014. "Tweets and Trades: The Information Content of Stock Microblogs." *European Financial Management* 20(5):926–57.

Stieglitz, Stefan and Linh Dang-Xuan. 2012. "Political Communication and Influence through Microblogging - An Empirical Analysis of Sentiment in Twitter Messages and Retweet Behavior." *Proceedings of the Annual Hawaii International Conference on System Sciences* 3500–3509.

Strub, Florian et al. 2017. "End-to-End Optimization of Goal-Driven and Visually Grounded Dialogue Systems." *IJCAI International Joint Conference on Artificial Intelligence* 13:2765–71.

Taboada, Maite, Julian Brooke, Milan Tofiloski, Kimberly Voll, and Manfred Stede. 2011. "Lexicon-Basedmethods for Sentiment Analysis." *Computational Linguistics* 37(2):267–307.

Tang, Duyu et al. 2014. "Learning Sentiment-Specific Word Embedding for Twitter Sentiment Classification." *52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014 - Proceedings of the Conference* 1:1555–65.

Tang, Duyu, Bing Qin, and Ting Liu. 2015. "Document Modeling with Gated Recurrent Neural Network for Sentiment Classification." *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing* (September):1422–32.

Turney, Peter D. 2001. "Thumbs up or Thumbs Down?" *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics - ACL '02* (July):417.

Vinet, Luc and Alexei Zhedanov. 2010. "A 'Missing' Family of Classical Orthogonal Polynomials." *Proceedings of the Fifth Annual Workshop on Computational Learning Theory - COLT '92* 144–52.

Vu, Ngoc Thang, Heike Adel, Pankaj Gupta, and Hinrich Schütze. 2016. "Combining Recurrent and Convolutional Neural Networks for Relation Classification." *CoRR* abs/1605.07333.

Wahid, Hairunnizam, Sanep Ahmad, Mohd Ali Mohd Nor, and Maryam Abd Rashid. 2017. "Prestasi Kecekapan Pengurusan Kewangan Dan Agihan Zakat: Perbandingan Antara Majlis Agama Islam Negeri Di Malaysia." *Jurnal Ekonomi Malaysia* 51(2):39–54.

Zeiler, Matthew D. 2012. “ADADELTA: An Adaptive Learning Rate Method.” *CoRR* abs/1212.5.

Zhang, Xue, Hauke Fuehres, and Peter A. Gloor. 2011. “Predicting Stock Market Indicators Through Twitter ‘I Hope It Is Not as Bad as I Fear.’” *Procedia - Social and Behavioral Sciences* 26:55–62.

Zhou, Xujuan, Xiaohui Tao, Jianming Yong, and Zhenyu Yang. 2013. “Sentiment Analysis on Tweets for Social Events.” *Proceedings of the 2013 IEEE 17th International Conference on Computer Supported Cooperative Work in Design, CSCWD 2013* (April 2010):557–62.

Zhu, Feng and Xiaoquan (Michael) Zhang. 2010. “Impact of Online Consumer Reviews on Sales: The Moderating Role of Product and Consumer Characteristics.” *Journal of Marketing* 74(2):133–48.