Multimed Tools Appl (2018) 77:32213–32242
https://doi.org/10.1007/s11042-018-6168-1

Deep neural network architecture for sentiment analysis and emotion identification of Twitter messages

Dario Stojanovski[1] · **Gjorgji Strezoski**[1] · **Gjorgji Madjarov**[1] · **Ivica Dimitrovski**[1] · **Ivan Chorbev**[1]

Received: 29 September 2017 / Revised: 28 March 2018 / Accepted: 21 May 2018 /
Published online: 18 June 2018

Abstract In the work presented in this paper, we showcase a deep learning system for sentiment analysis and emotion identification in Twitter messages. The system consists of a convolutional neural network used for extracting features from textual data and a classifier for which we experiment with several different classifying algorithms. We train the network using pre-trained word embeddings obtained by unsupervised learning on large text corpora and compare the effectiveness of the different word vectors for this task. We evaluate our system on 3-class sentiment analysis with datasets provided by the Sentiment analysis in Twitter task from the SemEval competition. Additionally, we explore the effectiveness of our approach for emotion identification, by using an automatically annotated dataset with 7 distinct emotions. Our architecture achieves comparable performances to state-of-the-art techniques in the field of sentiment analysis and improves results in the field of emotion identification on the test we use in our evaluation. Moreover, the paper presents several use case scenarios, depicting real-world usage of our architecture.

Keywords Twitter · Convolutional neural networks · Word embeddings · Sentiment analysis · Emotion identification

✉ Dario Stojanovski
stojanovski.dario@gmail.com

Gjorgji Strezoski
strezoski.g@gmail.com

Gjorgji Madjarov
gjorgji.madjarov@finki.ukim.mk

Ivica Dimitrovski
ivica.dimitrovski@finki.ukim.mk

Ivan Chorbev
ivan.chorbev@finki.ukim.mk

[1] Faculty of Computer Science and Engineering, Ss. Cyril and Methodius University, Skopje, Republic of Macedonia

1 Introduction

Language is the most common way of inter-human communication, one that has evolved into complex structures that are capable of conveying ideas and emotions. Understanding language poses no difficulty to humans in our everyday lives, but computers are still incapable to meaningfully comprehend language. Interacting with computers in natural language is a long quest of computer science and this area is popularly known as Natural Language Processing (NLP). There are many different specific research interests in this field, ranging from simple part-of-speech tagging to understanding and answering questions.

Social media platforms and microblogging services have attracted the attention of millions of users over the recent years. Twitter, Facebook and Instagram are just few examples of social media services that generate a massive amount of data. Twitter has over 316 million monthly active users and generates over 500 million messages on a daily basis.[1] The open nature of this microblogging service and the sheer size of the data being created, positions Twitter as the focal point of social media research and NLP. The Twitter community uses the platform in various ways and for different purposes, from daily chatter to spreading news. People express their opinions on a wide range of topics, political candidates, sport matches, movies or their experiences in using some products and services or just sharing their current mood.

Sentiment analysis is an area of NLP that is focused on understanding human emotion in text. This field received a lot of traction with the continuous growth in popularity of social media which provides many ways for people to express their thoughts. One of the first application of sentiment analysis was studying people's experience with certain products and services on review forums. However, determining sentiment in social media data is far more challenging as texts are usually short in length and contain very informal language. For example, Twitter limits the length of tweets to maximum 140 characters which inspires users to find new ways of expressing themselves, usually by using a lot of abbreviations, slang, URLs or with some of the Twitter specific terms such as hashtags and user mentions.

As Twitter provides easy access to the publicly available data, companies and marketing agencies are driven to collect and analyze this data to get a better insight into the market. As a result, sentiment analysis is of great interest to both, academia and industry looking to better current approaches to solving this challenge. Traditional NLP techniques for sentiment analysis are generally based on manually or automatically created lexicons [52] or hand-crafted features [34]. However, these approaches often produce over-specified and incomplete features and are both time-consuming to define and require extensive domain knowledge.

On the other hand, deep learning methods have recently been at the center of attention in many machine learning areas. Firstly, these methods have showed their potential in computer vision and audio processing and are just recently being applied in natural language processing. A significant advantage over previous approaches is the ability for automatic feature extraction which enables such models to learn more general representations, thus being more robust when applied to different domains.

In this paper, we continue on our previous work [47–49], employing a deep convolutional neural network for sentiment analysis in Twitter messages. The proposed architecture is based on the work of Kim [22] where the proposed model achieved highest performance in 4 out of 7 sentence classification tasks amongst which is a task with sentiment analysis

[1] http://about.twitter.com/company

classification. As opposed to this model, we employ several word embeddings and classifier, exploring deeper architectures with two non-linear layers and a SVM as last layer. We apply our proposed architecture to sentiment analysis in respect to three classes and emotion identification with 7 distinct emotions. For sentiment analysis, we use manually labeled dataset provided from the Sentiment analysis in Twitter task from the SemEval competition, while for emotion identification we utilize an automatically labeled set provided from the work done by Wang et al. [57]. Additionally, we present several use case scenarios in which we showcase how such system could be deployed. Our system architecture is applied for sentiment analysis of news-related tweets, where Twitter messages are retrieved and matched to corresponding news articles, potentially providing a way to get insight into the publics' reaction towards certain events. We also use our model to detect sentiment in geo-referenced tweets. Moreover, Twitter messages related to and posted during several games from the 2014 FIFA World Cup are analyzed in order to identify the expressed emotions.

The main contributions of this paper are three-fold:

- We evaluate different pre-trained word embeddings and classifiers in combination with the convolutional neural network and present the achieved performances.
- We apply our model on different domains including sentiment analysis and emotion identification and present the results.
- We present several use case scenarios in which our system architecture is employed.

The rest of the paper is organized as follows. Section 2 outlines current approaches in this field with emphasis on Twitter sentiment analysis and deep learning methods. In Section 3, we present in detail our proposed architecture along with the necessary pre-processing of the messages. Section 4 presents the experimental setup used in this work, overview of the datasets on which we train and test our model and contains an extensive evaluation of the model, with different pre-trained word embeddings, network parameters and classifiers. Section 5 present the different use case scenarios. Finally, we give the concluding remarks in Section 6.

2 Related work

Sentiment analysis has long been of interest to many researchers and there are many proposed techniques in this field. Some of the work focuses on more coarse granularity, detecting sentiment on document level [61], while others on more fine-grained granularity, detecting sentiment towards different aspects [29]. Since tweets are limited in length, they usually refer to a single concept or aspect and can be approximated with sentence level sentiment analysis. In this context, we identified three different groups of approaches in the domain of sentiment analysis and emotion identification. The first group addresses the problem of sentiment analysis and emotion identification by using rule-based approaches. The second group addresses the problem by using feature-based approaches, while the third group addresses the problem by using deep learning approaches that do not depend on extensive manual feature engineering.

2.1 Rule-based approaches

There are several previous works that use rule-based approaches to address sentiment analysis [24, 52, 60]. In general, these approaches propose building a sentiment polarity lexicons, created manually by domain experts or automatically using dictionary or corpus-based

methods. Such lexicons consist of a list of words with a polarity score associated with them, indicating whether the word is positive, negative or neutral. Manual approaches used for creating such lexicons are laborious and time-consuming, but provide for best quality of the lexicon. Nevertheless, they are usually used in combination with automatic ones in order to detect errors in the generated lexicons.

Sintsova et al. [43] used the Amazon Mechanical Turk (AMT) to build a human-based lexicon. Using the annotators' emotionally labeled tweets, they constructed a linguistic resource for emotion classification. Their approach is able to capture up to 20 distinct fine-grained emotions. However, using lexicons alone, is often insufficient. Same words can be used in different contexts, imply different sentiment and can be negated. Additionally, a sentence can contain an opinion, but that opinion is not necessarily expressed with an opinion-related words. This is especially true, for the social media domain, as users express themselves differently from the formal structure of the given language.

2.2 Traditional machine learning approaches

This group addresses the problem of sentiment analysis and emotion identification by using traditional machine learning approaches. These techniques use hand-crafted features and machine learning classifiers such as support vector machines, Naive Bayes classifier, logistic regression, multilayer perceptron, etc. These types of approaches require extensive feature engineering and domain knowledge. Defining appropriate features requires knowledge of how users are using Twitter and the language they use to express their mood and emotions.

Mohammad et al. [31] proposed a system which achieved best performance on classifying tweets polarity in the SemEval 2013 competition. They train a SVM classifier with linear kernel with extensive hand-crafted features. Some of the features are: word n-grams, character n-grams, whether a word is written in upper case, the number of occurrences of each Part-of-Speech tags, number of hashtags, number of contiguous sequences of exclamation and question marks and whether the last token is one of them, the presence of emoticons, the number of negated contexts, the number of elongated words etc. Additionally, they utilized three manually constructed sentiment lexicons where each word is mapped to a sentiment polarity score. The scores for each token, part-of-speech tag, hashtag and all-cap words are used to determine total count of words with score greater than zero, the total sentiment score, the maximal score and the score of the last token with score greater than zero. Agarwal et al. [1] defined around 100 features and explored the effectiveness of different feature combinations using SVM as a classifier. Additionally, they proposed using a tree kernel to generate a tree representation of the tweet and comparing two trees using the Partial Tree kernel.

Go et al. [13] proposed a system that use tweets with emoticons for distant supervised learning. They use different combinations of traditional machine learning approaches (Naive Bayes, maximum entropy, and support vector machines) and unigrams, bigrams, unigrams and bigrams, and unigrams with part of speech tags as features in the analysis.

Kouloumpis et al. [25] investigate the utility of linguistic features for detecting the sentiment of Twitter messages. They evaluate the usefulness of existing lexical resources as well as features that capture information about the informal and creative language used in microblogging. Pak and Paroubek [33] present a method for an automatic collection of a corpus that can be used to train a sentiment classifier. As a classifier they use Naive Bayes that uses N-gram and POS-tags as features.

Similar approaches in terms of feature engineering and used classifiers for the emotion identification problem are presented in [2, 12, 36, 39, 57].

2.3 Deep learning approaches

On the other hand, deep learning methods do not depend on extensive manual feature engineering and extract the features automatically. This is an advantage over other approaches, as such methods are more robust and easier to generalize to other domains. Generally, there are three deep learning techniques used for sentiment analysis, Recursive Neural Networks (RNN) [44], Long Short Term Memory (LSTM) [18] networks [27] and Convolutional Neural Networks (CNN) [5, 6, 10, 21, 22, 41]. However there are other proposed approaches. In the work of Zhou et al. [64] Restricted Boltzmann Machines have been used to evaluate the effectiveness of using labeled and unlabeled data. Tang et al. [54] propose a hybrid method between a deep learning and feature-based model.

2.3.1 Recursive neural networks

Socher et al. [44] introduced a Recursive Neural Tensor Network that maps phrases through word embeddings and a parse tree. Afterwards, vectors for higher nodes in the tree are computed and a tensor-based composition function for all nodes is used. The method pushed state-of-the-art results on fine-grained and positive/negative sentiment classification of movie reviews. This approach was also used for aspect specific sentiment analysis in the work of [26]. However, RNTNs depend on the syntactic structure of the text as input which needs to be generated beforehand.

2.3.2 Convolutional neural networks

CNNs on the other hand, do not depend on any additional input aside from the sentence or tweet itself. They have the advantage of inherently taking into account the ordering of the words and by using word embeddings they encompass syntactic and semantic meaning of words. Our work is based on the work of Kim [22] where a CNN with multiple filters was proposed for several sentence classification tasks, one of which is sentiment analysis in movie reviews. Severyn et al. [41] used the same approach for Twitter sentiment analysis, but used distant supervision to generate word embeddings. Similar approach was presented in the work of Chintala [5] where a deeper neural architecture was used. The authors proposed using a feed-forward neural network with linear and hyperbolic tangent non-linearity instead of a softmax classifier. In our work, we apply similar architecture, but we apply it to the more challenging domain of Twitter sentiment analysis. In [10], the authors explored the effectiveness of using character-level embeddings instead of just using word-level ones. Kalchbrenner et al. [21] proposed a Dynamic Convolutional Neural Network for modeling sentences. However, these papers do not report F1 scores achieved with their approaches and we can't fully compare to the presented performances. Tang et al. [54] proposed a hybrid system, where hand-crafted features are used in combination with distantly supervised generated word embeddings.

2.3.3 Long short term memory

LSTMs are widely used in NLP and as such are the basis of many approaches for sentiment analysis as well [27, 37, 56]. They take as input word embeddings and generate hidden states in a sequential manner where a given hidden state is dependent on the previous one. This allows these models to model long-range dependencies. Additionally, many approaches attempt to use a CNN followed by a LSTM [58], or train them jointly and use output from

both networks as the final representation [50], thus being able to model both, short-range and long-range dependencies.

Recently, Wang et al. [59] proposed to use an attention-based LSTM architecture to handle aspect-level sentiment analysis, while Kokkinos and Potamianos [23] use structural attention. However, their method depends on syntactic parse trees as input which are not easy to obtain with high quality for Twitter messages. Radford et al. [38] showed that training a language model in an unsupervised manner can also be used for sentiment analysis. They train the language model on Amazon product reviews and consequently use this representation to train a supervised sentiment classifier. Interestingly enough, they achieve state-of-the-art results and further observe that a single feature of the language model representation is responsible for detecting sentiment.

3 System architecture

In this paper we present a deep learning system for sentiment analysis and emotion identification on Twitter messages. The system consists of two parts, a feature extractor and a classifier. For the feature extraction part, we use a convolutional neural network, while for the classifier in this paper, we evaluate the performance of several different machine learning classifying algorithms. We present results produced using a softmax and a SVM classifier. Additionally, we report results obtained by applying a whole feed-forward neural network with several layers of non-linearity. Our system does not rely on any hand-crafted features, thus making it more robust and adaptable to several domains. An overview of the model we propose is given in Fig. 1. The implementation is in Python with the popular Theano framework [3, 4]. The phases of our architecture can be summarized as follows:

- Pre-processing on the tweets (pre-processing layer).
- Mapping each token of a tweet to a corresponding word embedding (embedding layer).
- Applying convolutional operation on the concatenated word embeddings for each of the defined filters (convolutional layer).
- Max-over-time pooling on the output features obtained from the convolutional layer (pooling layer).
- Classification of the fixed sized vector obtained form the max-over-time pooling layer (classification layer).

3.1 Pre-processing

Social media texts are written in informal language and contain many abbreviations, slang, emoticons, URLs etc. Twitter messages are no exception. Additionally, tweets are limited in length to 140 characters, thus forcing users to find new and often unpredictable ways of communication to enable to fully express themselves in this limited space. Twitter also enables users to mention other users in their tweets and to label the tweet in a way by supplying a suitable hashtag. Determining the expressed sentiment or emotion in these circumstances is very challenging. In order to clean the tweets of unnecessary information, we apply some pre-processing steps. The same cleaning was applied for both, sentiment analysis and emotion identification. Additional pre-processing was used for the task of emotion identification during the preparation of the dataset.

As URLs and HTML entities do not contribute to the expressed sentiment in any way we remove all such appearances. Additionally, all punctuation is removed with the exception

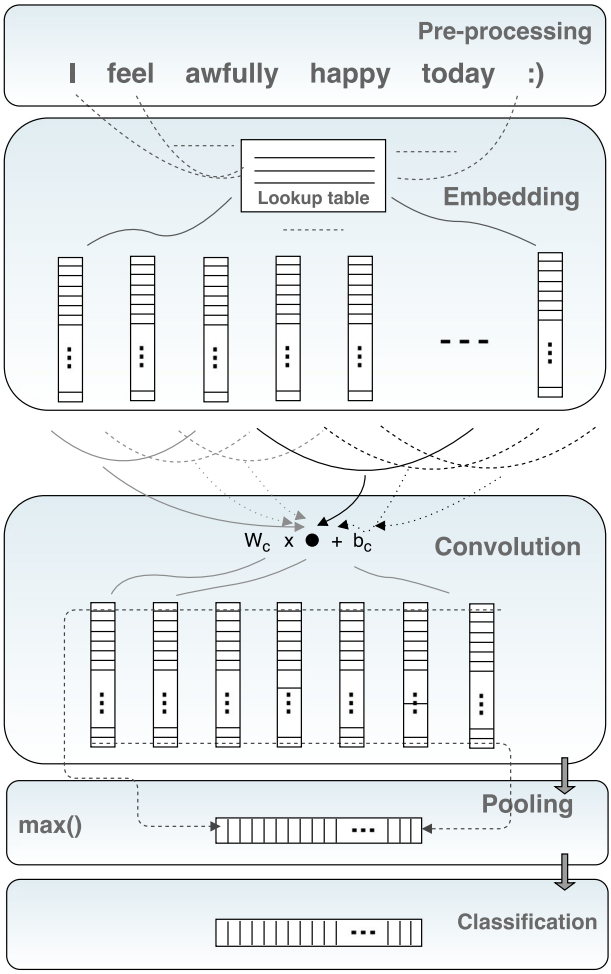

Fig. 1 Deep neural network architecture

of question and exclamation marks. Moreover, we keep emoticons, as one of the strongest indicators of the expressed sentiment. Twitter specifics such as hashtags are kept in the original form, unlike user mentions which are completely removed. Moreover, we lowercase all words. In previous work [48], we tried to replace abbreviations with their actual meaning by using a dictionary of social media abbreviations. However, we didn't observe improved performance and we contribute this mainly to the fact that suitable word embeddings are learned for the abbreviations in any event.

The final pre-processing step is the shortening of elongated words. It is common on social media sites for users to elongate words in order to emphasize certain aspects of their statement. However, the length of such words can vary greatly, but the same intention is being conveyed in either way. A simple example is a tweet containing the word "haaapy" and another one containing "haaaaapy". As we do not want to differentiate between such tokens, we shorten each elongated word to a maximum of three character repetitions.

Table 1 Details about the word embeddings used for the lookup table

	Dimension	Corpora	Vocabulary	Source
SSWE	50	10M	137K	Tweets
word2vec	300	100B	3M	GoogleNews
GloVe Crawl	300	840B	2.2M	Common Crawl
GloVe Twitter	200	20B	1.2M	Tweets

Corpora size is expressed in token count with the exception of SSWE where only the number of tweets is provided

3.2 Embedding layer

One of the most challenging parts of any machine learning task and consequently NLP, is finding a suitable feature representation for the dataset being used. Representing words and sentences or tweets in our case is especially challenging due to the enormous variations in human language. In our work, we represent a tweet as a concatenation of the word representations. Traditional NLP methods treat words as indexes, otherwise known as the one-hot vector representation. The representation is a large vector with zeros and a single one value. Having such approach, several issues arise. These vectors do not scale well as the vocabulary grows. Moreover, there is no semantic relationship between the words.

As a result, neural language models have been used to generate word representations that will overcome these shortcomings [30, 35, 55]. These so called word embeddings are learned in an unsupervised manner over large text corpora and project word representations to a lower dimensional space. The main intuition behind the idea is that given a large enough text corpus, a given word will always be in the vicinity of words syntactically and semantically similar to it and the meaning of the word can be inferred from its neighborhood. In this way, we end up with scalable feature vectors that encode syntactic and semantic regularities of the words.

Each word or token of the input tweet to our system is mapped to the corresponding word embedding. Word embeddings are stored in a lookup table that is generated before the training of the neural network. One way of generating the lookup table is by randomly initializing the word embeddings. However, previous work in this field [10, 22] report better performance with pre-trained word embeddings. There are several already available pre-trained word embeddings that can be utilized for this task. In this case, random initialization is used only for words that are missing from the lookup table. Apart from the widely popular word2vec[2] [30], we also evaluate the GloVe word embeddings[3] [35] and the Sentiment Specific Word Embeddings (SSWE) [55]. Details of the word embeddings we used in this research are presented in Table 1. Table 2 depicts the number of tokens that were found in the lookup table for each of the word embeddings for the different datasets used in this work.

The dimension represents the length of the feature vector while corpora presents the total number of tokens in the dataset used to train the word embeddings with the exception in the case of SSWE where only the number of tweets is reported. Vocabulary represents the number of unique words. It is important to note that this does not represent the total number

[2] https://code.google.com/p/word2vec/

[3] http://nlp.stanford.edu/projects/glove/

Table 2 Dataset dictionaries and number of matched tokens

	SSWE	word2vec	GloVe Crawl	GloVe Twitter	Dataset
sent	19196	17757	22642	22499	27990
emot 1K	6071	5601	6293	6328	7012
emot 10K	11684	10484	12411	12705	15003

of unique words in the dataset, but the word count of the available word embeddings. The final column shows what is the source of the dataset used to pre-train the word embeddings.

There are several issues with using word embeddings for the task of sentiment analysis in Twitter messages. As we already mentioned tweets are written in informal language while many of the available word embeddings are trained on corpora with more formal language. As a result, we expect that there will be more missing tokens and the representations might not be meaningful in this context. Consequently, we utilize word embeddings trained on Twitter data in order to overcome this issue. Additionally, word embeddings are trained in an unsupervised manner and as such there is no sentiment information encoded in the representation. Words that appear in similar context such as "good" and "bad" are neighboring words based on cosine similarity. In [41, 55] this issue was tackled by training embeddings in a distant supervised manner with emoticon labeled tweets. Word embeddings from the work of Tang et al. [55] (SSWE) are publicly available. However, the size of the corpus used for training is significantly smaller in comparison to other available word embeddings.

In our work, we approach this problem differently than that in the work of [55]. We use pre-trained word embeddings and update them during network training using back-propagation. The intuition is that by back-propagating the classification errors, sentiment regularities are encoded into the word representations. This is showcased in the work of Kim [22] where after network training, "good" was no longer similar to "bad". Additionally, this is a fitting way of building more meaningful representations for words that are not present in the vocabulary. Kim [22] suggests to use a range of $[-a, a]$ for the missing words where a is set so that the randomly initialized words have the same variance as the pre-trained ones. In our case, we set a to a value of 0.25.

3.3 Convolutional layer

For the feature extraction part of our architecture we use a convolutional neural network with multiple filters with varying window sizes. This research mainly bases on the work of Kim [22] where the proposed model achieved state-of-the-art performance on 4 out of 7 text classification tasks. Dealing with variable sized sentences is inherently built into CNNs with pooling operation and they additionally take into consideration the ordering of the words and the context each word appears in. Unlike application of CNNs in image understanding where usually there are multiple levels of convolutional layers, in the context of language understanding often one layer is sufficient and captures expressive features.

Let's consider a tweet t with length of n tokens. Because the filters in the convolutional layer are applied in a sliding window manner, we need to apply appropriate padding at the beginning and the end of the tweet. Padding length is defined as $h/2$ where h is the window size of the filter. Prior to network training we construct a lookup table where each word is represented with appropriate feature representation. The lookup table can be defined as $L \in R^{k \times |V|}$, where k is the dimension of the word vectors and V is a vocabulary of the words in the lookup table. Each word and token in the tweet is mapped to the corresponding

word embeddings and is projected to a vector $w_i \in R^k$. As a result a tweet is represented as a concatenation of the word embeddings as in (1) and is fed to the convolutional layer of the deep neural network:

$$x = \{w_1, w_2, ...w_n\} \tag{1}$$

The convolutional layer applies multiple filters with varying window sizes h on the tweet and produces feature maps as a result. For this domain, we choose to use window sizes of 3, 4 and 5 as tweets are limited to 140 characters and such sizes would be sufficient. Feature maps are obtained by repeated application of a function across sub-regions or windows of the entire tweet. In other words, we convolute a linear filter with the input tweet, add a bias term and finally apply a non-linear function. For each filter a weight matrix $w_c \in R^{h \times hk}$ and a bias term b_c are learned, where h is the number of hidden units in the convolutional layer. The convolutional operation can be formally expressed as:

$$x'_i = f(w_c \cdot x_{i:i+h-1} + b_c \tag{2}$$

where $x_{i:i+h-1}$ is the concatenation of word vectors from position i to position $i + h - 1$, while $f(\cdot$ is an activation function. For the purposes of this work, we have experimented with the hyperbolic tangent (3) or the rectified linear activation function (4). The weight matrix is used to extract local features around each word window.

$$f(x = tanh(x = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{3}$$

$$f(x = \begin{cases} 0 & \text{for } x < 0 \\ x, & \text{for } x \geq 0 \end{cases} \tag{4}$$

3.4 Pooling layer

The output of the convolutional layer is fed into a max-over-time pooling layer. This layer extracts the most important features for each feature map and additionally it handles the problem with varying tweet lengths. It outputs a fixed size vector which is necessary if we are to use a traditional classifying algorithm. The pooling operation is applied on the feature map x' computed from the convolution operation (5)

$$x' = max\{x'_1, x'_2 ...x'_{n-h+1}\} \tag{5}$$

The size of the output vector is correlated to the number of hidden units in the convolutional layer and is a hyper-parameter which is pre-defined. The outputs of the max-over-time pooling operation for each of the feature maps are concatenated and this represents the final output of the convolutional and max-over-time pooling layer. We train the network using stochastic gradient descent over shuffled mini-batches of size 50 with the Adadelta update rule [63].

3.5 Classification layer

The last part of our architecture is the classification layer (classifier) which determines whether a tweet is positive, negative or neutral for the sentiment analysis task or one of the 7 distinct emotions in the emotion identification task. In this research, we have experimented with several classifying algorithms and we present and compare the performances we achieved with each of them. Algorithms we have experimented with are a softmax regression classifier, SVM and different varieties of a feed forward neural network.

The softmax regression classifier generalizes logistic regression to a classification problem where the class label can take on more than two possible values. The softmax regression layer of our architecture gives the probability distribution over the labels. Given a test input x, softmax regression estimates the probability that $p(y = j|x)$ for each value of $j = 1,...,k$ where k is the number of classes. In other words, it estimates the probability of the class label taking on each of the k different possible values. The output is a k dimensional vector whose elements sum to 1 giving k estimated probabilities.

To our knowledge, utilizing a convolutional neural network in conjunction with SVM as a replacement for the more common approach of CNNs with a softmax regression classifier has not been explored in the context of NLP. However, there are several examples of such systems in the field of image processing. Some works extract features with a CNN and use them as input to the SVM classifier [19, 51], while others use a SVM-like layer as the top layer of the neural network [53]. In our work we borrow the approach used by Tang [53] as we need to backpropagate the errors during network training which is needed to update the word embeddings. Linear SVMs are binary classifiers thus we need to extend them for our multiclass problem with the *one-vs-all* approach. As a result k different linear SVMs are trained independently where one class is a positive and all of the rest are negative cases. In this case, SVMs generate predictions in the same way as a softmax regression classifier does, the difference being in their objectives parameterized by their weight matrices. Our softmax layer, maximizes log-likelihood while the SVM layer finds maximum margin between data points of different classes by computing the hinge loss. Tang [53] used a L2-SVM which minimizes the squared hinge loss:

$$\min \frac{1}{2} \quad ^T \quad + K \sum_{n=1}^{N} max(1 - \quad ^T x_n t_n \;\; 0 \;^2 \tag{6}$$

where N is the number of training examples and t_n is the class label where $t_n \in \{-1 \;\; 1\}$. The prediction of a class label is done as shown in (7).

$$arg \max_t (W^T x)t \tag{7}$$

Finally, we explore the effectiveness of a feed-forward neural network with several linear or non-linear layers and a softmax or SVM layer as a top layer of the network. Collobert et al. [6] propose using a CNN with a linear layer and a non-linear layer for which they chose a hard hyperbolic tangent function. Additionally, we explore the case where the first linear layer is replaced with a sigmoid activated layer. The first layer is represented as:

$$x^1 = f(\quad ^1 \dot{x}' + b^1 \tag{8}$$

where f is either a sigmoid or linear activation function, while the second layer

$$x^2 = tanh(\quad ^2 \dot{x}^1 + b^2 \tag{9}$$

is a hard hyperbolic tangent activated layer. Multiple layers of non-linearity can be used to model more complex representations of the data. The final layer of the network is again a softmax or a SVM layer like the ones presented before.

Deep neural architectures have a lot of parameters that are to be learned by the model. As a result these models suffer from overfitting which is most commonly tackled by employing dropout regularization [16]. What dropout regularization does is during the network training process some of the many hidden units and connections between them are randomly dropped (set to zero). The proportion of units that are to be dropped is a hyper-parameter and some suggestions of what are most efficient values of this parameter are reported in [45].

4 Experiments and results

In this work, we present the produced performances on the sentiment analysis dataset provided by the SemEval competition and the emotion identification dataset provided from the work of Wang et al. [57]. We explore several different aspects of our architecture, namely how different pre-trained word embeddings and different classifiers affect the accuracy and F1 score our network achieves. The macro-F1 score is computed for negative and positive classes for sentiment analysis as this is the metric used in the Twitter sentiment analysis track of the SemEval competition.

Our model has several parameters that need to be set. Kim [22] reports the values used in his research and we reuse some of the parameter values. The network is trained using mini-batches of size 50, l_2 constraint of 3 and filters with window sizes of 3, 4 and 5. We experimented with several different values for the other parameters of the network. Learning decay was set to 0.98. In the case where a feed-forward neural network is used as a classifier, the dropouts for the layers are 0.7 and 0.5 respectively for the two hidden layers. The number of hidden units in the convolutional layer was set to 100.

For the purpose of this research, we have experimented with several network layouts and several different network parameters. One aspect of evaluation is whether the activation function in the convolutional layer should be a hyperbolic tangent or rectified linear activation function. The scores in the following tables are with a hyperbolic tangent activation function as we observe better results in comparison to the rectified linear activation function. Moreover, we have evaluated several different classifying algorithms as a final layer of our deep neural network. As final layers, we chose a softmax regression classifier and a SVM and we also explored how we can combine them with linear or sigmoid activated layer or with a hyperbolic tangent layer in between them.

We evaluate based on F1 score which is computed as $F1 = \frac{2*P*R}{P+R}$ where P and R are precision and recall respectively. Precision is defined as $P = \frac{TP}{TP+FP}$, while Recall as $R = \frac{TP}{TP+FN}$, where TP denotes true positives, FP false positives and FN false negatives.

4.1 Sentiment analysis

The experimental part of our research consists of applying our model to sentiment analysis and emotion identification in the Twitter domain. Both tasks are a classification problem and therefore we require a labeled training set. We define the problem of sentiment analysis in our case as a three-way classification problem where the three labels are positive, negative or neutral sentiment. We train and test our model for this task using the benchmark sets from the SemEval challenge, specifically the Sentiment Analysis in Twitter task. All sets are manually labeled by the organizers of the challenge. However, we were not able to download all of the tweets provided by the competition. Mainly this is because of changed privacy settings of the tweet or the user that posted the tweet or simply because the tweet is deleted. The validation set was also provided by the SemEval competition which also supplied the three different test sets. Each test set is from different years of the competition. In the 2015 edition of the SemEval challenge, competitors were allowed to freely extend their datasets with additional data. We used manually labeled tweets[4] which were collected in respect to 4 main topics. Again, we were not able to download all of the provided tweets, having

[4] http://www.sananalytics.com/lab/twitter-sentiment/

Table 3 Distribution of labels on the training, development and test sets

	Positive	Negative	Neutral
Train	3307	1569	5843
Dev	464	261	593
TW13	1572	601	1640
TW14	982	202	669
TW15	1038	365	987

downloaded 3034 tweets where tweets marked as irrelevant were also removed. Details are provided in Table 3.

Tables 4, 5 and 6 present the achieved performance with different word embedding and classifier combinations. The values are the F1 scores for the Twitter 2013, 2014 and 2015 test sets respectively. We can observe that word2vec word embeddings almost consistently outperform SSWE, despite the fact that SSWE are trained on Twitter data and fine-tuned with distant supervision. This stems from the fact that word2vec embeddings are trained on significantly more data and have bigger dimensionality. It is interesting to note that using these vectors with a sigmoid activated first layer in the feed-forward neural network lowers the F1 score in comparison to other classifiers which is not the case with the other embeddings. GloVe word embeddings on the other hand, have proven to be more effective in our experimental evaluation. These word embeddings produce higher scores in comparison to the previous representations. The best results were obtained by using a sigmoid and hyperbolic tangent activated feed-forward neural network with GloVe Crawl word embeddings. However, it would be interesting to see if using GloVe Twitter embeddings with equal dimensionality to the GloVe Crawl embeddings would boost performances even higher as most F1 scores with GloVe word vectors differ by a narrow margin.

In our experimental evaluation, we observe that using a SVM layer does not improve performances. It actually lowered them in comparison to a softmax regression classifier. We don't report results with a SVM layer as the top layer of the network, with previous combination of linear and non-linear layers, as the performances with such network layout is significantly lower in comparison to the other presented F1 scores. Adding more layers to the network, whether it's a linear or non-linear, or a combination of them almost consistently improved performances using all of the aforementioned word embeddings. Considering our best performing combination of word embedding and a classifier, we achieve top performances for the Twitter 2015 test set amongst the competition. The F1 score with the same setup for the Twitter 2013 and Twitter 2014 test sets are 69.11 and 71.11 respectively. On the Twitter 2013 test set best performance was achieved with a two layer network where

Table 4 F1 scores on the Twitter 2013 test set for sentiment analysis

	SSWE	word2vec	GloVe Crawl	GloVe Twitter
Softmax	66.16	67.68	68.67	66.74
SVM	63.18	65.59	65.95	66.15
Linear + Softmax	65.63	67.17	69.48	68.63
Sigmoid + Softmax	65.45	63.17	68.18	68.56
Linear + Tanh + Softmax	65.79	67.58	70.15	66.58
Sigmoid + Tanh + Softmax	66.24	61.72	69.11	69.55

Table 5 F1 scores on the Twitter 2014 test set for sentiment analysis

	SSWE	word2vec	GloVe Crawl	GloVe Twitter
Softmax	67.46	66.6	69.25	66.97
SVM	64.99	65.66	68.3	67.14
Linear + Softmax	66.99	67.23	69.88	68.17
Sigmoid + Softmax	68.25	62.53	69.36	70.31
Linear + Tanh + Softmax	67.22	66.67	70.79	67.72
Sigmoid + Tanh + Softmax	66.41	64	71.11	70.28

the first layer is a linear one. However, the best performing combination on the other two datasets lags behind only by 1%.

We compare the performance of our model on the Twitter 2015 test against results reported by several top-scoring teams at SemEval 2015. The best performing system based on hand-crafted features [15] scored 64 84, but failed to match on the other two test sets, suggesting that it may be over-fitted. The system proposed in [9], which is a hybrid approach using various hand-crafted and deep learning features, achieves best performance on Twitter 2013 and 2014, 72 8 and 74 42 respectively, but fails on Twitter 2015, scoring 63 73. [41] report consistently high results across all test sets, 72.79, 73.36 and 64.59, but they train their model's embeddings in an unsupervised manner on a huge tweets dataset. Furthermore, they improve upon by fine-tuning them in a semi-supervised manner on large-scale distantly labeled tweets dataset, which was not available to us and would require significant computing power.

4.2 Emotion identification

Acquiring a dataset for the emotion identification task is more challenging as there are not any manually labeled publicly available datasets. Therefore, we are forced to resort to an automatically labeled dataset which is far larger in quantity, but unfortunately, it would not be on the same quality level as a manually annotated one. We build our work on a dataset provided from the work of Wang et al. [57]. They define 7 distinct human emotions (*love*, *joy*, *surprise*, *anger*, *sadness*, *fear*, *thankfulness*). For each of these emotions they have extrapolated set of keywords and their lexical variants to represent a single human emotion. Then, the Twitter API was queried for tweets containing any of the keywords in the form of a hashtag.

Table 6 F1 scores on the Twitter 2015 test set for sentiment analysis

	SSWE	word2vec	GloVe Crawl	GloVe Twitter
Softmax	58.06	60.18	61.14	61.47
SVM	54.63	58.81	61.8	60.48
Linear + Softmax	57.73	60.93	62.22	61.67
Sigmoid + Softmax	58.85	56.58	62.45	62.99
Linear + Tanh + Softmax	58.38	61.03	62.02	61.06
Sigmoid + Tanh + Softmax	57.88	57.22	64.88	62.72

Table 7 Emotion identification results

	SSWE	word2vec	GloVe Crawl	GloVe Twitter
Softmax	47.4%	48.95%	50.75%	51.95%
SVM	48.35%	50.6%	50.3%	50.25%
Linear + Softmax	48.3%	47.4%	49.4%	50.25%
Sigmoid + Softmax	43.25%	47.1%	47.65%	49.8%
Linear + Tanh + Softmax	45.7%	45.2%	47.1%	49.05%
Sigmoid + Tanh + Softmax	43.25%	44.7%	45.75%	48.2%

The authors applied additional heuristics to filter the set. Tweets containing URLs, quotations, more than 3 hashtags were removed in addition to non-English tweets, retweets and tweets where the emotion related hashtag was not at the end of the tweet. Moreover, the quality of the dataset was manually evaluated by randomly sampling a small portion of the dataset for manual inspection by human annotators. The overlap between automatically and manually assigned labels is 95.08% for the development and 93.16% for the test set, testifying for the quality of the data. However, many of the tweets were not available for download for reasons we mentioned before. Out of the 1991184 training samples, 247798 for development and 250000 for test we were only able to retrieve 1347959, 168003 and 169114 respectively. Due to technical limitations, we did not utilized the full capacity of the dataset. We tested our model with 2000 Twitter messages while for the development set we used 1000 messages. The distribution of emotion labels in our reduced dataset matches the one of the original dataset. We applied the same heuristics that are pertaining to the removal of the hashtags indicative of the emotion from the Twitter message.

For our emotion identification experiments, we borrow the approach we used for sentiment analysis. Again, we evaluate different classifiers and word embeddings and additionally, different network parameters. Table 7 shows the obtained performances on the emotionally labeled dataset where the values this time present the accuracy the model achieved on the test set. The training set used for the results presented in this table contains 1K samples. Table 8 presents precision, recall and F1 score for each of the seven emotion classes with the 10K samples training set.

For the task of emotion identification, again we can observe that word2vec word embeddings almost consistently outperform SSWE, while the same can be stated for GloVe Crawl with respect to word2vec and GloVe Twitter with respect to GloVe Crawl. The GloVe word embeddings used in our research are trained on informal data, whether it is from crawling the Web or from Twitter messages, and this is most probably the main reason for the

Table 8 Precision, recall and F1 score for each emotion label

	Precision	Recall	F1 score
Joy	65.31	70.52	67.82
Sadness	61.51	53.96	57.49
Anger	57.56	73.93	64.73
Love	42.29	33.79	37.56
Fear	37.5	11.76	17.91
Thankfulness	48.78	62.5	54.79
Surprise	/	/	/

increased performances in comparison with the word2vec embeddings. It is interesting to note here that for this task adding more layers to the network almost consistently lowered performances. In the case where more layers were used in combination with a SVM top layer the network achieved significantly worse performances and as a result we do not report on these accuracies. The top performance for this task is produced with GloVe Twitter embeddings and a single softmax regression layer. We note again, that using GloVe Twitter with a dimensionality of 300 could additionally improve performances.

Our model outperforms the accuracy that is reported in the work of Wang et al. [57] on both training sets. Their model achieves 43.41% accuracy with a 1K training set and 52.92% with a 10K training set. On the other hand, our deep convolutional neural network architecture obtains accuracy of 51.95% and 58.84% respectively which is a considerable improvement. The three most common labels in the training set are *joy* (28%), *sadness* (24%) and *anger* (22%) and as a result the highest F1 score is obtained for these three emotion labels. Both precision and recall are high in comparison to the other labels. For the less present emotion labels, *love* (12%) and *fear* (5%) precision is relatively high, while recall is significantly lower. *Thankfulness* (5%) on the other hand has a relatively high recall as well. Unfortunately, the training set is consisted of only 1% of samples with *surprise* emotion label. As a result, our model didn't correctly predicted any of the test samples with this particular emotion class.

5 Applications

5.1 Sentiment in news-related tweets

Exponential growth and worldwide reach of social media has transformed the way people consume and spread news. Today news are being disseminated on social media platforms long before they reach traditional news sources. Twitter is on the forefront of this transition as has been demonstrated on many occasion in the past few years. This has sparked great interest in the scientific community. There are many works done to enable matching news to relevant Twitter messages [8, 11, 14, 17, 28, 40, 42, 46].

In this use case, we showcase a system for sentiment analysis of social media conversations that are matched to relevant news articles. News are crawled from popular news sources such as BBC and The Guardian, while tweets are retrieved by utilizing the Twitter Streaming API. An overview of the system is depicted in Fig. 2. The proposed system enables to get insight into how the public reacts to certain events mentioned in news articles by observing the number of positive, negative and neutral received tweets.

We crawl news articles from BBC and The Guardian from several topics including worldwide news, business, sport and technology news. News articles are kept for a period of 24 hours and are presented in the list of all news. After the completion of that time interval, the news article is no longer matched to Twitter messages and is removed from the news list. The system is implemented in Python and for web page crawling we use the Scrapy library.[5] News crawling is executed periodically on every 10 minutes. During the interval of 24 hours each of the news articles is matched to corresponding tweets posted during the same time period. In order to accomplish this we extract the 5 most important keywords from each article using the traditional TF-IDF scheme. These keywords are then supplied

[5] http://scrapy.org/

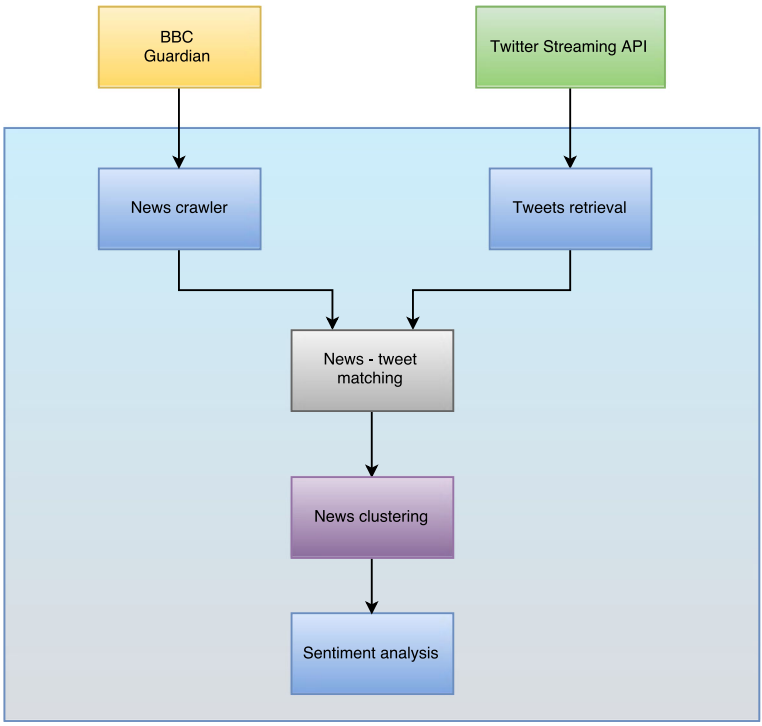

Fig. 2 System architecture of sentiment analysis on social conversations related to news articles

to the news retrieval module which queries the Twitter Streaming API for tweets containing these keywords. So as to narrow down the search space, we only download tweets that contain any combination of two out of the five keywords for every tweet. Unlike news crawling, tweet retrieval is continuous and retrieves tweets matching the aforementioned criteria.

However, the collected tweets are not considered matched at this stage as we apply additional filtering. The filtering process is also executed periodically on every 10 minutes. All of the tweets retrieved during those 10 minutes are compared against each of the currently active news articles. The comparison is computed based on simple cosine similarity between the TF-IDF vectors representing the articles and the tweets. Each tweet whose cosine similarity to certain news article is above a given similarity threshold is considered related to the news article. The threshold value is manually defined.

Upon news-tweet matching, we apply clustering on the tweets in order to group them into separate topics. The method used in this module of the system is borrowed from the work of Ifrim et al. [20] where tweets are hierarchically clustered and a headline is produced for each of the topic clusters. They compute tweet pairwise distances using cosine similarity as a metric and apply hierarchical clustering using the *fastcluster*[6] library. The main idea is that each cluster will represent a different topic related to the news article. In order not to predefine the number of clusters, hierarchical clustering is used and the dendogram is cut at value 0.5. Higher threshold value would result in looser clusters that can potentially contain

[6] http://danifold.net/fastcluster.html

tweets from different topics, while lower values would result in tighter clusters, but can lead to topic fragmentation, meaning that several clusters would represent the same topic. Determining the topic headline is done by selecting the first tweet, chronologically speaking as the topic headline. We select the top-50 clusters for presentation in the web interface of our system. All other tweets are grouped in a special "All" class which contains all of the remaining tweets.

For this use case, we use our model for sentiment analysis and apply it to tweets related to news articles. We classify each tweet in respect to three classes, positive, negative and neutral. As a result, we present the number of positive, negative and neutral tweets for each of the generated clusters with the corresponding headlines.

At first, the user is presented with a list of all of the crawled news articles. The list can be sorted by popularity or recency and contains information for the title, date and time of publication, shorten version of the content of the article and the number of matched Twitter messages.

The detail screen of the web interface presents the full version of the news article, the 5 most important keywords from both the content of the article and from the aggregated tweets in order to observe any correlation between them. More importantly, it presents the matched tweets clustered into separate topics and the number of positive, negative and neutral tweets with respect to each topic. It is interesting to note that many of the tweets are labeled as neutral as a lot of the tweets do not contain any personal opinion but only provide URL to the article and a copy of the article headline. Clicking on any of the headlines will expand the full list of tweets related to the appropriate cluster. We provide a demo of the system.[7]

5.2 Emotion identification in 2014 FIFA World Cup tweets

Global sporting events attract the attention of millions of users all around the world. The Olympics and the FIFA World Cup are arguably the most followed events and this is reflected in user activity on social media. During the 2014 FIFA World Cup, Twitter witnessed a staggering 672 million tweets related to the competition. Access to this enormous amount of data is available through the Twitter Streaming API and there have been several examples of sport related analysis on Twitter [7, 32, 62].

Sports games have big effect on the viewers and can evoke a variety of emotional reactions. Fans tend to express their emotions on social media for the duration of the game, thus giving us the opportunity to tap into the stream of tweets and perform analysis on the expressed sentiment. One could even put such a system in a real-world use case where emotional reactions can be used to detect potential violent outbursts on the sport fields which is a common sight on sport stadiums, especially on football (soccer) games.

We choose the 2014 FIFA World Cup as a base for our emotion identification analysis that is explained in details in [47]. We showcase our system on three games from the competition, Belgium vs. USA, Brazil vs. Germany and Brazil vs. Netherlands. The Belgium vs. USA game was chosen because of the fact that there were three goals scored in the overtime of the game, potentially causing strong emotional responses from the fans. Brazil was the host nation of the World Cup and therefore we chose there matches in the semi-final and for the third place. Brazil vs. Germany was one of the most interesting games in the competition and one with most scored goals.

[7] http://194.149.136.27/Paper/TwitterNews/home

The dataset was collected using the Twitter Streaming API which enables real-time access to tweets based on their geo-location, the user that posted them, whether they contain certain keywords or not etc. In our work, we only retrieved English tweets regardless of the location they were posted from. The API was queried for tweets containing the official game hashtag (i.e. #BRAGER) or general World Cup hashtags (i.e. #WorldCup2014).

The original emotion labeled training set has an imbalanced distribution of emotions. Emotions such as thankfulness or surprise are rarely predicted as a result of this imbalance and our system is bias towards more common labels in the training set. To counteract this issue we train our model with a training set with a balanced distribution of emotion labels by selectively sampling training examples. The size of the training set is 10000 samples and all labels are equally present in the dataset. Table 9 depicts examples of tweets from the FIFA World Cup classified with our model. In the remainder of this section we provide analysis of the emotional reactions we received during the three games. The following figures depict the distribution of emotions for the duration of the game. The x-axis shows the time in GMT, while the y-axis shows the number of tweets labeled with a certain emotion.

It is obvious that we will observe spikes in user activity at and immediately after important and interesting events of the game. Understandably, goals are the biggest actuators of emotions which is apparent on the figures below. In our system, we track tweets from all around the world thus the potential bias towards some team will be less present. However, due to the reason that we only gather English tweets, we can assume that most of the tweets will come from English speaking countries and a bias for example can be expected towards USA in the Belgium vs. USA game. Additionally, fans traditionally are more inclined to support host nations, even more so when the host nation is Brazil which is very liked among fans from all around the world. Therefore, slight bias can be expected in the other two matches towards Brazil.

The quarter-final match between Belgium vs. USA was fairly uneventful during the regular time of the match (Fig. 3). Probably due to early enthusiasm among USA supporters, we can observe that positive emotions are more present, *love* dominates over other emotions and *joy* is close to *sadness*. The distribution evens out after the first few minutes and continuous unchanged to the end of the match. The first bigger increase in Twitter activity is after the end of the game, because fans were likely excited that the game went into overtime and the possibility of a penalty shootout which is always accompanied with a lot of tension and excitement. It is interesting to note that the first goal by Belgium scored in 93rd minute (21:56 GMT) actually received smaller attention than the end of the game. The second goal, again scored by Belgium in the 105th minute (22:08 GMT) was accompanied by a larger response, but both goals evoked similar emotional responses, where positive and negative emotions were equally present. However, the last goal scored by USA in the 107th minute of the match (22:10 GMT) received significantly larger attention with *love* dominating other emotions. This sentiment proves the predicted bias towards USA and continued on after the end of the game even though the USA didn't progress to the next phase of the competition.

The second game we chose for analysis, the semi-final between Brazil and Germany is probably the most interesting game of the whole 2014 FIFA World Cup. We have retrieved over 1.1 million tweets during this game, a substantially greater number of messages in comparison with the other games. Consequently, we received a constant high data stream, with tweets containing a mixture of different emotions. Because of the fact that the final score was 7-1 for Germany and the fact that Brazil conceded 4 goals in just 6 minutes, football fans were likely euphoric and were motivated to post on social media platforms. Moreover, it is likely that users that rarely post sport related tweets were also interested in this game

Table 9 Sample tweets from the World Cup set with the predicted emotion labels

Sadness	That Robben booking was ridiculous. Another example of home field advantage for Brazil #BRAvsNED
Joy	Looks like Brazil will lose horribly again. NICE! #WorldCup2014
Fear	Please don't let this be another massacre. #BRAvsNED

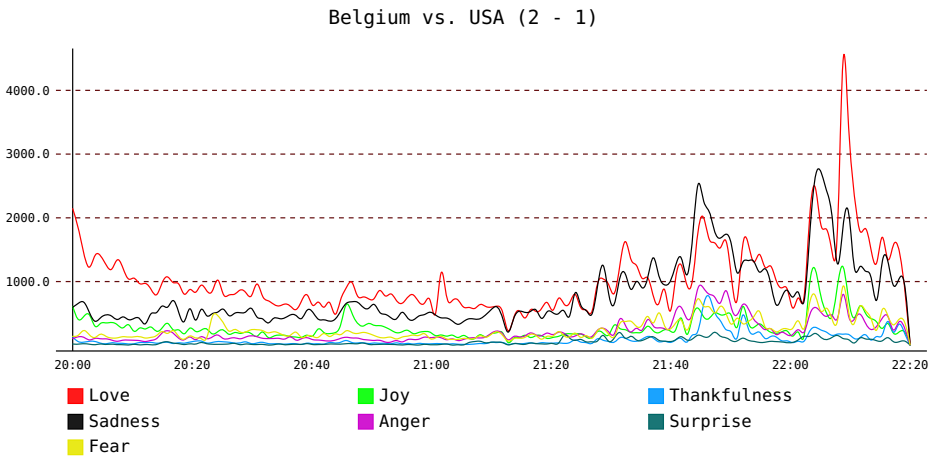

Fig. 3 Quarter-final: Belgium vs. USA

and joined the sport community on Twitter. The game started with a fairly even distribution of emotions which was interrupted by the first goal in the 11th minute (20:11 GMT). It is interesting to observe that this caused dominantly positive emotions such as *love* and *joy* in spite of the bias towards Brazil that we assumed. We contribute this to fans' excitement of an early goal and expectation of an open game with many chances and goals. However, sentiment shifted dramatically from the 23rd minute (20:23 GMT). Germany scored 4 quick goals (23rd minute 20:23 GMT, 24th minute - 20:24 GMT, 26th minute - 20:26 GMT, 29th minute - 20:29 GMT) and this caused a chaotic mixture of emotions among the fans which can be observed in Fig. 4. Throughout the rest of the game, *sadness* has become the dominant emotion with *anger* and *fear* also having a significant appearance. This is in line with our assumption of fans' inclination towards the host nation. The number of posted tweets remained constantly high with the exception during the half-time and in the second half up

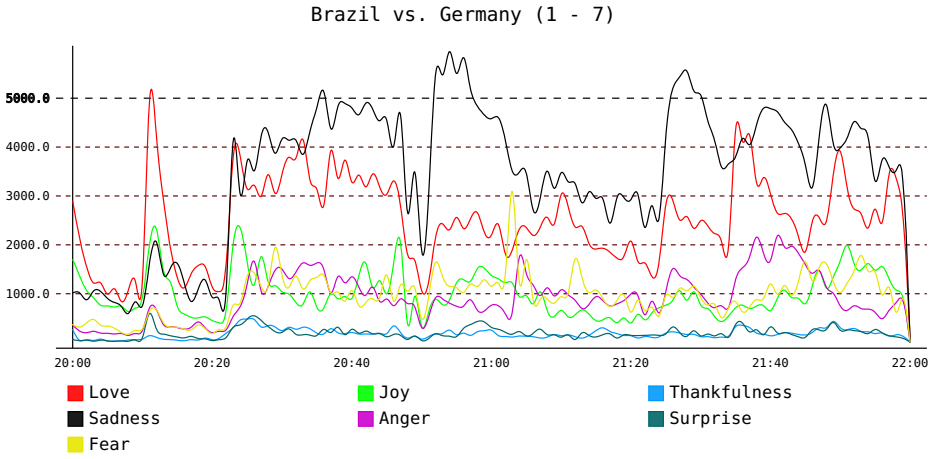

Fig. 4 Semi-final: Brazil vs. Germany

until the sixth goal in the 69th minute (21:24 GMT). Similar distribution of emotions can be observed from this moment on regardless of the fact that Brazil managed to score a goal in the 90th minute (21:53 GMT) which did not produced any noticeable increase in Twitter activity.

The last game is the match for the third place between Brazil and the Netherlands. The game started of with two goals by the Netherlands (3rd minute - 20:03 GMT, 17th minute - 20:17 GMT). These events mainly evoked negative emotions with *fear* being in the more present emotions. We contribute this to the fans being afraid of another high loss like the one from Germany. For the remainder of the game, there were not many notable events and this shows on Fig. 5 where the distribution of emotions evens out. Spikes in activity are present for the third goal by the Netherlands in the 91st minute (21:54 GMT) and another around 21:30 GMT which is probably noise from the dataset.

5.3 Sentiment in social hotspots

The last use case is in the field of Volunteered Geographic Information which has attracted a lot of attention since the explosion of geo-referenced data, especially on social media. Smartphones with built-in GPS sensors enable users to easily share their location. Twitter users are also sharing their current whereabouts and through the Twitter API access to tweets from certain geographical location is easy. In this section, a system for detecting social hotspots from the Twitter stream and applying sentiment analysis on the data is presented. We showcase our system on geo-data from New York. Details about the system are given in our work [49].

The presented system monitors Twitter streams for a defined geographic area and identifies social hotspots as they are emerging. The system is broken down to the following key components:

- Data retrieval - connects to the Twitter Streaming API and collects messages from a certain geographic area.
- Cluster generation - creates clusters or social hotspots from tweets retrieved in a limited time period.

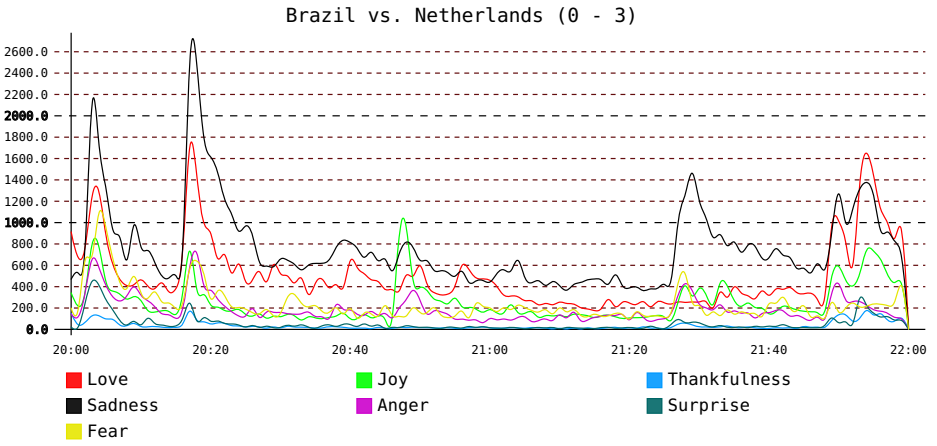

Fig. 5 3rd place: Brazil vs. Netherlands

- Sentiment analysis - analyses the sentiment of the tweets in the social hotspot clusters.

5.3.1 Data retrieval and overview

As with the previous use cases, we utilize the Twitter Streaming API to collect tweets. However, in this case we don't query the API for tweets containing certain keywords, but for tweets originating from a given geographic location defined by a set of spatial bounding boxes. The bounding box is defined by a set of longitude and latitude pairs. In order for a tweet to be geo-tagged, the user must explicitly give consent to share his location. Additionally, Twitter enables manual embedding of geographical places in posts. Places on Twitter have IDs, can be of several different types and are defined by a bounding box. However, since including such places in Twitter messages is manual, the user may not necessarily be located at that specific place.

We collect tweets from New York, USA. All tweets that contain specific location which is within the defined bounding box or have place embedding whose bounding box intersects with the one supplied to the API are retrieved by our system. Another problem with posts with manual place embeddings is that a significant portion refer to greater geographical locations such as cities. As a result, a retrieved tweet can actually be outside of the desired bounding box.

We collected tweets between February 22 and April 16 2015. We set the bounding box to the following longitude and latitude pairs: (-74, 40), (-73, 41). For the above mentioned period, we collected 4125542 Twitter messages, generated from a total of 226114 distinct users. Out of these, 3274724 messages contained exact coordinates, while the others had a place entity only, which were attached manually. However, we observed that among these tweets that also had place entities attached, a significant portion was outside of the defined bounding box. Upon filtering, the dataset was reduced to 2350739 messages. In the set collected, we observed that place entities generally are related to greater geographical areas. For example, places such as "Manhattan", "New York" or "Brooklyn" appear very often, as opposed to points of interests. Such places are insignificant to our analysis and have to be filtered out because they refer to a very broad area. Only 9119 tweets with embedded POIs that are within the defined bounding box were retrieved, while 269 messages have POIs outside of the bounding box.

5.3.2 Hotspot detection and sentiment

A social hotspot is a geographic POI which attracts the attention of many people in a limited period of time. In the context of Twitter, detecting social hotspots requires locating places with highly concentrated activity. In order to identify social hotspots from Twitter streams, clusters of geographically close tweets must be created. We have experimented with both, hierarchical agglomerative clustering and the DBSCAN algorithm. The generated clusters can be further analyzed in order to determine their relatedness with each other. However, we leave the development of such system for future work and we consider each generated cluster as a social hotspot.

The system presents the overall sentiment of a social hotspot which can even be used for recommending points of interests to users as it can provide them with feedback for the popularity of a social hotspot in real-time. In Fig. 6 a sentiment heatmap is depicted, where the red color represents negative, blue represent positive and green neutral clusters. The sentiment of the cluster is determined based on the sentiment of the tweets that make up the social hotspot. The most present sentiment in the tweets is the overall sentiment of the

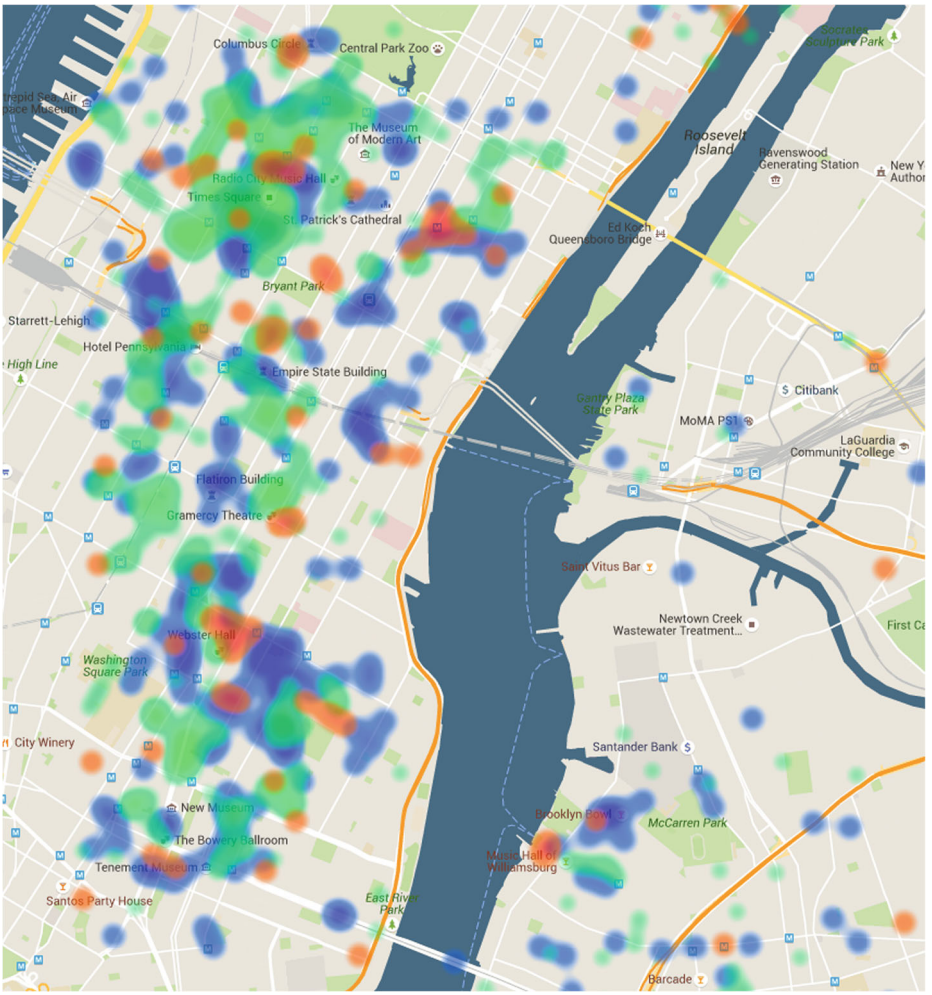

Fig. 6 Sentiment heatmap

cluster. From this map we can observe where is the atmosphere among Twitter users more positive and where is negative.

6 Conclusion

In this paper we present our approach to sentiment analysis and emotion identification in Twitter messages. The model consist of a convolutional neural network as a feature extractor form the tweets, while for the classifier, we evaluate the performance of different algorithms and their effect on the F1 score. The presented research also explores the effect different pre-trained word embeddings have on the achieved results. We test our approach on four pre-trained embeddings, namely SSWE, word2vec and GloVe trained on Common Crawl

and a tweets dataset. From our experiments, we observe that using a context relevant corpora for the pre-training is very important as GloVe vectors trained on Web data and Twitter messages produce better performances on both, sentiment analysis and emotion identification. Word2vec word embeddings almost consistently outperform SSWE, despite the fact that SSWE are trained on Twitter data and fine-tuned with distant supervision. We also evaluate different configurations of our proposed architecture. We observed that, for sentiment analysis, consistently high results are obtained with a two-layer feed-forward network on top of the convolutional layer, while the use of a SVM layer decrease the performances. For emotion identification however, the best results are obtained by employing only a softmax layer. In the case where more layers were used in combination with a SVM top layer the network achieved significantly worse performances. Our proposed architecture achieves comparable performance to the state-of-art methods in sentiment analysis, getting a F1 score of 64.88. For the task of emotion identification, we outperform previous work on our test set, achieving an accuracy of 58.84% using a training set with 10K samples. Additionally, we showcase three use case scenarios where we apply our model. We present how an emotion identification system can be utilized to detect fans' emotional response during sports matches. We provide analysis of emotion distribution on three games from the 2014 FIFA World Cup. Moreover, our sentiment analysis model is applied to detect the public reactions on Twitter towards current news articles for which we present a demo version of such system. Finally, we showcase a system for sentiment analysis of geo-referenced tweets for the New York area.

Acknowledgements We would like to acknowledge the support of the European Commission through the project MAESTRA Learning from Massive, Incompletely annotated, and Structured Data (Grant number ICT-2013-612944). Also, this work was partially financed by the Faculty of Computer Science and Engineering at the Ss. Cyril and Methodius University.

References

1. Agarwal A, Xie B, Vovsha I, Rambow O, Passonneau R (2011) Sentiment analysis of twitter data. In: Proceedings of the workshop on languages in social media, association for computational linguistics, Stroudsburg, PA, USA, LSM '11, pp 30–38. http://dl.acm.org/citation.cfm?id=2021109.2021114
2. Balabantaray R, Mohammad M, Sharma N (2012) Multi-class twitter emotion classification: a new approach. Int J Appl Inform Syst 4(1):48–53
3. Bastien F, Lamblin P, Pascanu R, Bergstra J, Goodfellow IJ, Bergeron A, Bouchard N, Bengio Y (2012) Theano: new features and speed improvements. Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop
4. Bergstra J, Breuleux O, Bastien F, Lamblin P, Pascanu R, Desjardins G, Turian J, Warde-Farley D, Bengio Y (2010) Theano: a CPU and GPU math expression compiler. In: Proceedings of the python for scientific computing conference (SciPy), oral presentation
5. Chintala S (2012) Sentiment analysis using neural architectures. New York University
6. Collobert R, Weston J, Bottou L, Karlen M, Kavukcuoglu K, Kuksa P (2011) Natural language processing (almost) from scratch. J Mach Learn Res 12:2493–2537. http://dl.acm.org/citation.cfm?id=1953048.2078186
7. Corney D, Martin C, Göker A (2014) Spot the ball: Detecting sports events on Twitter. In: Advances in information retrieval. Springer, pp 449–454
8. Demirsoz O, Ozcan R (2017) Classification of news-related tweets. J Inf Sci 43(4):509–524
9. Dong L, Wei F, Yin Y, Zhou M, Xu K (2015) Splusplus: a feature-rich two-stage classifier for sentiment analysis of tweets. In: Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015), pp 515–519

10. dos Santos C, Gatti M (2014) Deep convolutional neural networks for sentiment analysis of short texts. In: Proceedings of COLING 2014, the 25th international conference on computational linguistics: technical papers. Dublin City University and Association for Computational Linguistics, pp 69–78. http://aclweb.org/anthology/C14-1008
11. Freitas J, Ji H (2016) Identifying news from tweets. In: Proceedings of the first workshop on NLP and computational social science, pp 11–16
12. Ghazi D, Inkpen D, Szpakowicz S (2010) Hierarchical approach to emotion recognition and classification in texts. In: Advances in artificial intelligence. Springer, pp 40–50
13. Go A, Bhayani R, Huang L (2009) Twitter sentiment classification using distant supervision. CS224N Project Report. Stanford, pp 1–12
14. Guo W, Li H, Ji H, Diab MT (2013) Linking tweets to news: a framework to enrich short text data in social media. In: ACL (1). Citeseer, pp 239–249
15. Hagen M, Potthast M, Büchner M, Stein B (2015) Webis: an ensemble for twitter sentiment detection. In: Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015), pp 582–589
16. Hinton GE, Srivastava N, Krizhevsky A, Sutskever I, Salakhutdinov RR (2012) Improving neural networks by preventing co-adaptation of feature detectors. arXiv:12070580
17. Hoang-Vu TA, Bessa A, Barbosa L, Freire J (2014) Bridging vocabularies to link tweets and news. In: Seventeenth International workshop on the web and databases (WebDB 2014)
18. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780
19. Huang FJ, LeCun Y (2006) Large-scale learning with svm and convolutional for generic object categorization. In: 2006 IEEE Computer society conference on computer vision and pattern recognition, vol 1. IEEE, pp 284–291
20. Ifrim G, Shi B, Brigadir I (2014) Event detection in twitter using aggressive filtering and hierarchical tweet clustering. In: SNOW-DC@ WWW, pp 33–40
21. Kalchbrenner N, Grefenstette E, Blunsom P (2014) A convolutional neural network for modelling sentences. arXiv:14042188
22. Kim Y (2014) Convolutional neural networks for sentence classification. arXiv:14085882
23. Kokkinos F, Potamianos A (2017) Structural attention neural networks for improved sentiment analysis. arXiv:170101811
24. Kolchyna O, Souza TT, Treleaven P, Aste T (2015) Twitter sentiment analysis: Lexicon method, machine learning method and their combination. arXiv:150700955
25. Kouloumpis E, Wilson T, Moore J (2011) Twitter sentiment analysis: the good the bad and the omg! Icwsm 11:538–541
26. Lakkaraju H, Socher R, Manning C (2014) Aspect specific sentiment analysis using hierarchical deep learning. In: NIPS Workshop on deep learning and representation learning
27. Le P, Zuidema W (2015) Compositional distributional semantics with long short term memory. arXiv:150302510
28. Lin X, Gu Y, Zhang R, Fan J (2016) Linking news and tweets. In: Australasian database conference. Springer, pp 467–470
29. Lu B, Ott M, Cardie C, Tsou BK (2011) Multi-aspect sentiment analysis with topic models. In: 2011 IEEE 11th International conference on data mining workshops (ICDMW). IEEE, pp 81–88
30. Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J (2013) Distributed representations of words and phrases and their compositionality. In: Burges C, Bottou L, Welling M, Ghahramani Z, Weinberger K (eds) Advances in neural information processing systems, vol 26. Curran Associates, Inc., pp 3111–3119. http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf
31. Mohammad S, Kiritchenko S, Zhu X (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
32. Nichols J, Mahmud J, Drews C (2012) Summarizing sporting events using twitter. In: Proceedings of the 2012 ACM international conference on intelligent user interfaces. ACM, pp 189–198
33. Pak A, Paroubek P (2010) Twitter as a corpus for sentiment analysis and opinion mining. In: LREC, vol 10, pp 1320–1326
34. Pang B, Lee L (2008) Opinion mining and sentiment analysis. Found Trends Inform Retriev 2(1–2):1–135
35. Pennington J, Socher R, Manning C (2014) Glove: global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). Association for Computational Linguistics, Doha, pp 1532–1543. http://www.aclweb.org/anthology/D14-1162
36. Purver M, Battersby S (2012) Experimenting with distant supervision for emotion classification. In: Proceedings of the 13th conference of the European chapter of the association for computational linguistics. Association for Computational Linguistics, pp 482–491

37. Qian Q, Huang M, Lei J, Zhu X (2016) Linguistically regularized lstms for sentiment classification. arXiv:161103949
38. Radford A, Jozefowicz R, Sutskever I (2017) Learning to generate reviews and discovering sentiment. arXiv:170401444
39. Roberts K, Roach MA, Johnson J, Guthrie J, Harabagiu SM (2012) Empatweet: annotating and detecting emotions on Twitter. In: LREC, pp 3806–3813
40. Sankaranarayanan J, Samet H, Teitler BE, Lieberman MD, Sperling J (2009) Twitterstand: news in tweets. In: Proceedings of the 17th acm sigspatial international conference on advances in geographic information systems. ACM, pp 42–51
41. Severyn A, Moschitti A (2015) Twitter sentiment analysis with deep convolutional neural networks. In: Proceedings of the 38th International ACM SIGIR conference on research and development in information retrieval. ACM, pp 959–962
42. Shi B, Ifrim G, Hurley N (2014) Insight4news: connecting news to relevant social conversations. In: Machine Learning and knowledge discovery in databases. Springer, pp 473–476
43. Sintsova V, Musat CC, Pu Faltings P (2013) Fine-grained emotion recognition in olympic tweets based on human computation. In: 4th Workshop on computational approaches to subjectivity, sentiment and social media analysis, EPFL-CONF-197185
44. Socher R, Perelygin A, Wu JY, Chuang J, Manning CD, Ng AY, Potts C (2013) Recursive deep models for semantic compositionality over a sentiment treebank. In: Proceedings of the conference on empirical methods in natural language processing (EMNLP). Citeseer, pp 1631–1642
45. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res 15(1):1929–1958
46. Štajner T, Thomee B, Popescu AM, Pennacchiotti M, Jaimes A (2013) Automatic selection of social media responses to news. In: Proceedings of the 19th ACM SIGKDD international conference on knowledge discovery and data mining. ACM, pp 50–58
47. Stojanovski D, Strezoski G, Madjarov G, Dimitrovski I (2015) Emotion identification in fifa world cup tweets using convolutional neural network. In: 11th International conference on innovations in information technology (IIT), pp 52–57
48. Stojanovski D, Strezoski G, Madjarov G, Dimitrovski I (2015) Twitter sentiment analysis using deep convolutional neural network. In: Hybrid artificial intelligent systems. Springer, pp 726–737
49. Stojanovski D, Chorbev I, Dimitrovski I, Madjarov G (2016) Social networks vgi: Twitter sentiment analysis of social hotspots. In: European Handbook of crowdsourced geographic information, pp 223–235
50. Stojanovski D, Strezoski G, Madjarov G, Dimitrovski I (2016) Finki at semeval-2016 task 4: deep learning architecture for twitter sentiment analysis. In: Proceedings of the 10th International workshop on semantic evaluation (SemEval-2016), pp 149–154
51. Strezoski G, Stojanovski D, Dimitrovski I, Madjarov G (2015) Deep learning and support vector machine for effective plant identification. ICT Innovations 2015. Web Proceedings ISSN null, pp 41–50
52. Taboada M, Brooke J, Tofiloski M, Voll K, Stede M (2011) Lexicon-based methods for sentiment analysis. Comput Linguist 37(2):267–307
53. Tang Y (2013) Deep learning using linear support vector machines. arXiv:13060239
54. Tang D, Wei F, Qin B, Liu T, Zhou M (2014) Coooolll: a deep learning system for twitter sentiment classification. In: Proceedings of the 8th international workshop on semantic evaluation (SemEval 2014), pp 208–212
55. Tang D, Wei F, Yang N, Zhou M, Liu T, Qin B (2014) Learning sentiment-specific word embedding for twitter sentiment classification. In: Proceedings of the 52nd Annual meeting of the association for computational linguistics (volume 1: long papers). Association for Computational Linguistics, pp 1555–1565. http://aclweb.org/anthology/P14-1146
56. Tang D, Qin B, Feng X, Liu T (2015) Effective lstms for target-dependent sentiment classification. arXiv:151201100
57. Wang W, Chen L, Thirunarayan K, Sheth AP (2012) Harnessing twitter "big data" for automatic emotion identification. In: 2012 International Conference on and 2012 international confernece on social computing (SocialCom) privacy, security, risk and trust (PASSAT). IEEE, pp 587–592
58. Wang J, Yu LC, Lai KR, Zhang X (2016) Dimensional sentiment analysis using a regional cnn-lstm model. In: Proceedings of the 54th annual meeting of the association for computational linguistics (volume 2: short papers), vol 2, pp 225–230
59. Wang Y, Huang M, Zhao L et al (2016) Attention-based lstm for aspect-level sentiment classification. In: Proceedings of the 2016 conference on empirical methods in natural language processing, pp 606–615
60. Wilson T, Wiebe J, Hoffmann P (2005) Recognizing contextual polarity in phrase-level sentiment analysis. In: Proceedings of the conference on human language technology and empirical methods in natural language processing. Association for Computational Linguistics, pp 347–354

61. Yessenalina A, Yue Y, Cardie C (2010) Multi-level structured models for document-level sentiment classification. In: Proceedings of the 2010 conference on empirical methods in natural language processing. Association for Computational Linguistics, pp 1046–1056
62. Yu Y, Wang X (2015) World cup 2014 in the Twitter world: a big data analysis of sentiments in us sports fans' tweets. Comput Hum Behav 48:392–400
63. Zeiler MD (2012) Adadelta: An adaptive learning rate method. arXiv:12125701
64. Zhou S, Chen Q, Wang X (2010) Active deep networks for semi-supervised sentiment classification. In: Proceedings of the 23rd international conference on computational linguistics: posters. Association for Computational Linguistics, pp 1515–1523

Dario Stojanovski is a PhD student at the Center for Information and Language Processing at LMU Munich under the mentorship of Dr. Alexander Fraser. His PhD topic is on Neural Machine Translation, focusing on domain adaptation techniques. He finished his bachelor's and master's degree at the Facility of Computer Science and Engineering at the Ss. Cyril and Methodius University in Skopje, Macedonia. During his master's studies, he worked as a junior researcher on the Maestra project. He worked on sentiment analysis and emotion identification in Twitter messages and on various applicative use-cases. His master's thesis, was supervised by Dr. Gjorgji Madjarov. His work on Twitter sentiment analysis was presented and published at international conferences and book chapters.

Gjorgji Strezoski is a PhD candidate at the Informatics Institute of the University of Amsterdam. Currently he is working on multimedia analytics and artificial intelligence in the art domain with deep nets. Graduated at the Ss. Cyril and Methodius university at the Faculty for Computer Science and Engineering in Skopje in 2014 and obtained a Masters degree in Software Engineering at the same institution in 2015. In November 2016 he started his PhD at the University of Amsterdam on Visual Analytics for paintings. His current research interests are Computer Vision with Multitask Learning, Deep Generative Models and Information Visualization.

Gjorgji Madjarov received his bachelor and master degrees in computer science, automation and electrical engineering from the Faculty of Electrical Engineering and Information Technology, University "Ss. Cyril and Methodius" in Skopje, R. of Macedonia in 2007 and 2009, respectively. In 2012, he received his PhD in Computer Science from the Faculty of Computer Science and Engineering, University "Ss. Cyril and Methodius". He is currently an Associate Professor at the Faculty of Computer Science and Engineering, University "Ss. Cyril and Methodius". His main research interests are on multi-class, multi-label and hierarchical multi-label classification and ranking, and image processing, His wider fields of interest, also include supervised and unsupervised learning, computer vision, data mining, and pattern recognition.

Ivica Dimitrovski received his bachelor and master degrees in computer science, automation and electrical engineering from the Faculty of Electrical Engineering and Information Technology, University "Ss. Cyril and Methodius" in Skopje, R. of Macedonia in 2005 and 2008, respectively. In 2011, he received his PhD in Computer Science from the same institution. He is currently an Associate Professor at the Faculty of Computer Science and Engineering, University "Ss. Cyril and Methodius". His main research interests are on computer vision, natural language processing and machine learning. He is also a co-founder of the company GRID system. The company main focus is on natural language processing, web development and mobile application development.

Ivan Chorbev is an Associate professor at the Faculty of Computer Science and Engineering. He has completed his PhD and MSc at the "Ss. Cyril and Methodius" University – Skopje, Faculty of Electrical Engineering and Information Technologies. He has published more than 90 papers in several conferences, journals, books in the area of heuristic optimization algorithms, combinatorial optimization, e-medicine, heuristic algorithms in medicine, knowledge extraction, machine learning, medical data mining, telemedicine, assistive technologies, software engineering etc. Member of the IEEE and the IEEE Computer Society.