# Emotion Identification in Twitter Messages for Smart City Applications

**Dario Stojanovski, Gjorgji Strezoski, Gjorgji Madjarov and Ivica Dimitrovski**

## 1 Introduction

Social media have gained increasing popularity over the recent years with the number of users on social networks and microblogging platforms growing rapidly. As of 2016, Twitter has over 310 million monthly active users and generates over 500 million messages per day.[1] Twitter messages are also known as tweets and have 140 character limitation. Tweets contain informal language, users use a lot of abbreviations, URLs, emoticons and Twitter specific symbols, such as hashtags and targets (user mentions).

Emotion identification and sentiment analysis have recently spiked the interest of both, academia and industry with the exponential growth of social media. Detecting users' reaction towards certain products and services can provide valuable insight for companies offering them. Additionally, it can be used to get information of the public opinion against different topics and events and many other potential use-cases. As a result there are many proposed methods for solving this task.

Twitter emotion analysis can have many potential smart city applications. In previous work [1], we presented a use-case scenario where we apply our emotion

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

D. Stojanovski (✉) · G. Strezoski · G. Madjarov · I. Dimitrovski
Faculty of Computer Science and Engineering, Ss. Cyril and Methodius University,
Rugjer Boshkovikj 16, 1000 Skopje, Republic of Macedonia
e-mail: stojanovski.dario@gmail.com

G. Strezoski
e-mail: strezoski.g@gmail.com

G. Madjarov
e-mail: gjorgji.madjarov@finki.ukim.mk

I. Dimitrovski
e-mail: ivica.dimitrovski@finki.ukim.mk



L. Ismail and L. Zhang (eds.), *Information Innovation Technology in Smart Cities*,
https://doi.org/10.1007/978-981-10-1741-4_10

identification model to Twitter messages from the 2014 FIFA World Cup in Brazil. We provided an extensive analysis of emotional distribution detected for the duration of the matches being considered in our work. Emotion identification in sports tweets can be utilized for different applications. An emotion identification system can detect if a game has caused a lot of anger or other negative emotions amongst the fans, so the official organizers can be warned to take extra security measures after and during the match. Additionally, results from such system can be used to identify future potentially critical matches in terms of security.

Another potential smart city application of Twitter emotion analysis is in relation to local public services. Several works study how Twitter is used in this context. Sobaci and Karkin [2] present how mayors in Turkey use the social network to offer better public service. In Agostino [3] and Flores and Rezende [4] on the other hand, they analyze how citizens use Twitter to engage in public life and decisions. They showcase that the platform is heavily used by both, the government and the citizens, and can be utilized to improve communication between them in order to offer better services.

In the work presented in this paper, we showcase a deep learning system for emotion identification in Twitter messages. For this purpose, we use a convolutional neural network with multiple filters and varying window sizes which has not been adequately studied for the task of emotion detection. The approach is founded on the work of Kim [5] that reported state-of-the-art results in 4 out of 7 sentence classification tasks. We leverage pre-trained word embeddings, obtained by unsupervised learning on a large set of Twitter messages [6]. The network is trained on automatically labeled tweets in respect to 7 emotions. Additionally, we present a use-case scenario for smart city applications that leverages local government related messages. We apply our model for emotion identification in tweets related to local government projects and municipality issues.

The rest of the paper is organized as follows. Section 2 outlines current approaches to emotion identification. In Sect. 3, we present the proposed system architecture consisted of a convolutional neural network and the necessary pre-training. We give a detailed overview of the used dataset, the conducted experiments and the achieved results in Sect. 4. In Sect. 5, we present a use-case where we apply our system for emotion identification in tweets related to local government. Finally, we conclude our work in Sect. 6.

## 2 Related Work

There has been a lot of work done in the field of emotion identification. Current approaches mainly are based on unigrams, bigrams, Part-of-Speech tags and other hand-crafted features and machine learning algorithms such as Support Vector Machines (SVM), Naive Bayes and maximum entropy.

Appropriate labeling of tweets with corresponding emotions still poses a challenge in the field. Roberts et al. [7] used a manually annotated set of tweets with 7 labels for emotion. Their approach for classification consisted of hand-crafted features such as unigrams, bigrams, indicators of exclamation and question marks, WordNet hypernyms and several other features and a SVM classifier. Balabantaray et al. [8] also used manually labeled tweets and developed a system for emotion classification using hand-crafted features such as Part-of-Speech tags, unigrams, bigrams and others and the Word-net Affect emotion lexicon. Their work presents an extensive analysis of the effect different combinations of features have on performance.

Purver and Battersby [9] and Wang et al. [10] on the other hand, use distant supervision for automatic emotion annotation. By using well-known indicators of emotional content they were able to create noisily labeled datasets. Tweets were annotated by the presence of emotion-related hashtags. Wang et al. [10] applied additional heuristics to improve on the quality of the acquired dataset. They only considered messages where the emotion-related hashtag is at the end of the tweet, removed messages containing URLs and quotations and discarded tweets with more than 3 hashtags, non-English messages and retweets. Furthermore, the quality of the dataset was evaluated by randomly sampling a small number of tweets for manual inspection by two annotators. They received 95.08% precision on the development and 93.16% on the test set, where a tweet was manually checked whether the assigned label is relevant to the conveyed emotion. The dataset is publicly available[2] and we build our work on portion of the data.

Sintsova et al. [11] 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.

Unlike the previously depicted approaches that use extensive feature engineering which can be both, time-consuming and produce over-specified and incomplete features, our approach is based on a deep learning technique. Deep learning approaches handle the feature extraction task automatically, potentially providing for more robust and adaptable models. Most of the current work focuses on utilizing convolutional neural networks. Collobert et al. [12] proposed a unified neural network architecture that can be applied to various Natural Language Processing (NLP) tasks including sentiment analysis. dos Santos and Gatti [13] proposed a CNN for identifying sentiment analysis by exploiting character-level, word-level and sentence-level information. Kim [5] on the other hand, proposed an approach for sentence classification including sentiment analysis using a CNN with multiple filters and feature maps. This work also showed that continuously updating pre-trained word embeddings provides for better performance. In our work, we build on the aforementioned deep learning techniques. However, these approaches have been used for sentiment analysis, which is limited to classifying tweets with three labels, positive, negative and neutral. In this paper, we use convolutional neural network for a finer grained classification into 7 emotions.

[2] http://knoesis.org/projects/emotion.

## 3 System Architecture

The approach we used for emotion identification in this work is a convolutional neural network architecture which is depicted in detail in Fig. 1. The model is consisted of a simple network with one convolutional layer and a softmax output layer. Each token in a tweet is represented by a word embedding or word representation generated by a neural language model [6]. These features are fed to the convolutional layer of the network. The proposed system is not dependent on hand-crafted features and manually created lexicons. Consequently, the approach is more robust than traditional NLP techniques and more adaptable when applied to different tasks and domains.

### 3.1 Pre-training

In order to clean noise from the tweets, we applied several pre-processing steps. We replaced each occurrence of a user mention with a generic token and lowercased all

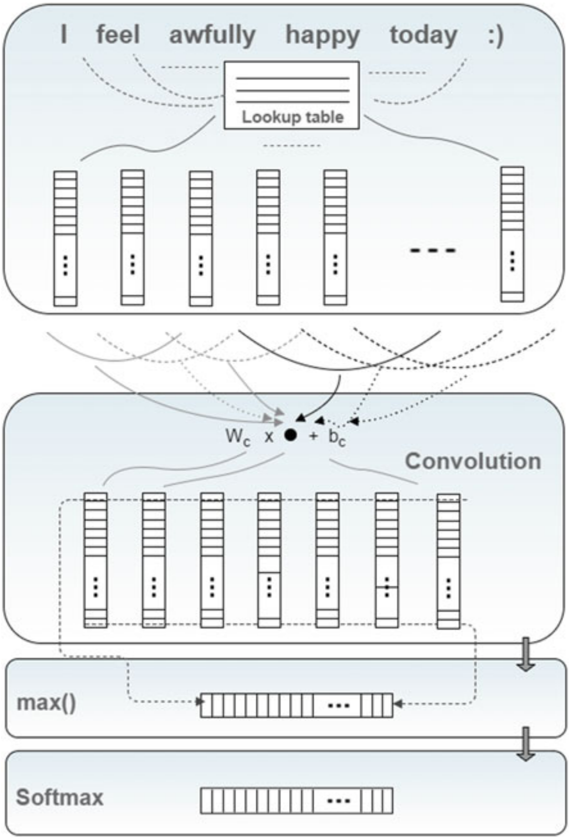


**Fig. 1** Convolutional neural network architecture

words. Additionally, we removed all HTML entities and punctuation except for exclamation and question marks and stripped hashtag symbols, but we kept emoticons. Moreover, all elongated words were shortened to maximum 3 character repetitions.

We leverage word representations or embeddings, learned on large corpus of unlabeled textual data using neural language models. The word embeddings are a continuous representation of the words themselves. These embeddings capture syntactic and semantic regularities of words and have recently been used in many tasks in NLP. Our system works by first, constructing a lookup table, where each word is mapped to an appropriate feature vector or word embedding. In this work, we do not do the pre-training of word embeddings ourselves as there are several already available ones, which are used to initialize the lookup table.

Since we are dealing with Twitter messages, which usually contain a lot of informal language, slang and abbreviations, using word embeddings trained on corpus where more formal language is used may not be suitable. Using word vectors trained on Twitter data is more fitting in our task as we assume there will be less missing tokens in the lookup table and the representations will be more meaningful in this context. Therefore, we leverage 200 dimensional GloVe embeddings [6] trained on 2 billion tweets (20 billion tokens). For words that are not present in the vocabulary of word vectors, we use random initialization.

However, since the training is done in an unsupervised manner, there is no sentiment or emotion regularities encoded in the embeddings. As a result, words such as “bad” and “good”, that likely appeared in similar context in the corpus are neighboring words based on cosine similarity. We use available word embeddings and by back-propagation, during network training, update them in order to adapt to the specific task at hand. The intuition behind this approach is that by back-propagating the classification errors, emotion regularities are encoded into the word representations. Upon finishing network training, “good” was no longer one of the most similar words to “bad” and vice versa. Additionally, this approach enables for building a more meaningful representation for words that are not present in the lookup table and for which random initialization is used.

### 3.2 Convolutional Neural Network

In this work, we utilize a convolutional neural network for classification of tweets into 7 emotion classes. Our approach is based on the work of Kim [5] which was used for different sentence classification tasks including sentiment analysis. CNNs with pooling operation deal naturally with variable length sentences and also, to some extent, take into account the ordering of the words and the context each word appears in. For simplicity, we consider that each tweet represents one sentence.

The network is trained by mapping tweets to an appropriate feature representation and supplying them to the convolutional layer. Each word or token of an input tweet, with the appropriate padding at the beginning and end of it, is mapped to an appropriate word representation. Padding length is defined as $h/2$ where $h$ is the window size of the filter. Words are mapped from the aforementioned lookup table $L \in R^{k \times |V|}$, where $k$ is the dimension of the word vectors and $V$ is a vocabulary of the words in the lookup table. Each word or token is projected to a vector $w_i \in R^k$. After the mapping, a tweet is represented as a concatenation of the word embeddings

$$x = \{w_1, w_2, \ldots, w_n\}.$$

The obtained feature representation of the tweet is then supplied to the convolutional layer. In this step, we apply multiple filters with varying windows sizes $h$. We use rectified linear units in the convolutional layer and windows of size 3, 4 and 5. As tweets are short texts with limited character count, having such window sizes is adequate and using larger ones would not be beneficial. Filters are applied to every possible window of words in the tweet and a feature map is produced as a result. For each of the filters, a weight matrix $W_c$ and a bias term $b_c$ are learned. The weight matrix is used to extract local features around each word window. The convolution operation can be formally expressed as:

$$x_i' = f(W_c \cdot x_{i:i+h-1} + b_c),$$

where $f(\cdot)$ is the activation function and $x_{i:i+h-1}$ is the concatenation of word vectors from position $i$ to position $i + h - 1$. The generated feature map is then passed through a max-over-time pooling layer:

$$x' = \max\{x_1', x_2', \ldots, x_{n-h+1}'\},$$

which outputs a fixed sized vector where the size is a hyper-parameter to be determined by the user. In our case, we set the size of this vector to 100 and this hyper-parameter corresponds to the number of hidden units in the convolutional layer. By doing so, we extract the most important features for each feature map.

The output of the pooling operation for each of the convolution operations with varying window sizes is concatenated. Predictions are generated using a softmax regression classifier. The concatenated features from the max-over-time pooling layer are passed to a fully connected softmax layer whose output is the probability distribution over the labels.

Deep neural networks suffer from overfitting due to the high number of parameters that need to be learned. In order to counteract this issue, we use dropout regularization which essentially randomly drops a portion of hidden units (sets to zero) during training. As a result, the network prevents co-adaption between the hidden units. The proportion of units to be dropped is hyper-parameter to be determined by the user. The network is trained using stochastic gradient descent over shuffled mini-batches.

# 4 Experiments

## 4.1 Dataset

In order to train our model, we utilize an already available annotated set provided in the work of Wang et al. [10]. Tweets in this dataset are annotated with 7 basic emotions *love*, *joy*, *surprise*, *anger*, *sadness*, *fear* and *thankfulness*. The dataset was generated by extrapolating a set of keywords of the 7 basic human emotions and their lexical variants to represent a single category of human emotions. Then, they queried the Twitter API for tweets containing any of the keywords in the form of a hashtag. The dataset is not related to any specific topic or domain.

However, due to Twitter privacy policy, only the IDs of the tweets were available for download, not the content itself. Using the Twitter API, we collected the annotated messages, but because of changed privacy settings or deletion, a significant portion of the messages was not available. Out of 1,991,184 tweets for training, 247,798 for development and 250,000 for test, we were able to retrieve 1,347,959, 168,003 and 169,114, respectively. Nonetheless, this is still a representative dataset. Moreover, the distribution of emotions in the tweets was similar to that of the original set. We applied the same heuristics that are pertaining to the removal of the hashtags indicative of the emotion from the Twitter message.

## 4.2 Experimental Setup and Results

We reused several parameters which were used in the work of Kim [5], mini-batch size of 50, $l_2$ constraint of 3, rectified linear units for the convolutional layer and filter windows of 3, 4 and 5. We set the learning rate to 0.02 and the dropout parameter to 0.7. These parameters, along with the decision to use rectified linear units over hyperbolic tangent was done by doing a grid search using the 1000 samples training set.

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. Both sets were generated by randomly sampling from the retrieved tweets. We trained the model with 1000 and 10,000 training samples, in order to observe the gains from a larger training set. Employing the above mentioned parameters we were able to achieve 50.12% with 1000 training samples and 55.77% with 10,000. Wang et al. [10] reported 43.41 and 52.92% respectively. Our model achieves higher accuracy in both cases on our reduced set.

Results for each label are presented in Table 1. For the three most popular emotions, *joy* (28%), *sadness* (24%) and *anger* (22%), we observe the highest F1-score of 64.95, 55.48 and 58.71% respectively. Both precision and recall are

**Table 1** Precision, recall and F1 score for each emotion label

| Emotion | Precision (%) | Recall (%) | F1 (%) |
|---|---|---|---|
| Joy | 59.59 | 71.38 | 64.95 |
| Sadness | 51.06 | 60.74 | 55.48 |
| Anger | 59.28 | 58.16 | 58.71 |
| Love | 44.68 | 34.29 | 38.8 |
| Fear | 52.78 | 16.52 | 25.16 |
| Thankfulness | 71.01 | 41.89 | 52.69 |
| Fear | 0 | 0 | 0 |

relatively high in comparison to the other emotions. Precision and recall for *joy* are 59.59 and 71.38%, 51.06 and 60.74% for *sadness* and 59.28 and 58.16% for *anger*. For the less popular ones, *love* (12%), *fear* (5%), *thankfulness* (5%) precision is relatively high, 44.68, 52.78 and 71.01% respectively, but recall is significantly lower compared with the top 3 emotions, 34.29, 16.52 and 41.89% respectively. The F1-score for each of these emotions are 38.8, 25.16 and 52.69% accordingly. The imbalance of class distribution in the dataset, leads to a classifier that will rarely classify a sample with uncommon labels. Since *surprise* accounts for only 1% of the training data, on the randomly sampled test set in our work, the classifier did not classify correctly any test example with *surprise*.

## 5 Emotion Identification in Local Government Tweets

In this paper, we present a use-case of emotion identification for local government purposes. We outline a system that can potentially be used by local governments to get real-time feedback from the local community in relation to ongoing and proposed projects. A showcase of the proposed system is presented in Fig. 2. We implemented the system for several major cities in USA such as New York, Los Angeles, San Francisco etc. The system is implemented in the Django web framework.

First, we manually identify official local government Twitter accounts for each corresponding city. This also includes the official accounts of the city's mayor and accounts for the 311 number which provide access to information and non-emergency municipal services. For example, for New York, we retrieve tweets from *@nycgov*, *@NYCMayorsOffice*, *@BilldeBlasio* and *@nyc311*. For the retrieval of Twitter messages, we utilize the Twitter Streaming API which enables access to tweets posted in real-time. Using the Streaming API, we collect tweets posted by these users. All messages mentioning any of the defined accounts are retrieved as well. In this way, we do not depend solely on what local governments tweet, but what the local community is concerned with too.

However, it doesn't necessarily mean that all tweets related to local government will mention some of the official accounts. As a result, in order to provide for bigger coverage, we also download geo-referenced tweets posted within some of the cities


**Fig. 2** Main page overview of our proposed system

under observation. This is also enabled by the Twitter Streaming API. Supplying the API with a bounding box defined by a set of coordinates, it will return every tweet posted within those boundaries. Tweets can be geo-referenced by either containing exact coordinates or by being embedded with a Place object. Places are Twitter specific objects which are defined by their name and bounding box among other information. They can relate to different geographic objects, from narrow areas such as points of interest to wider ones, such as entire cities. Tweets containing Place objects whose bounding box intersects with those of our interest are returned by the API.

Then, these tweets are compared against tweets posted by the local government accounts in order to identify which ones are related to actual local government issues. The matching between tweets is done by computing the cosine similarity of the TF-IDF representation of the Twitter messages. We can potentially, narrow the search space by only retrieving tweets containing at least two keywords from local government posted tweets. In this way, we get a high probability of getting related tweets, although additional filtering would still be required.

The system enables overview of local government related Twitter activity for New York, Boston, Washington DC, Detroit, Los Angeles and San Francisco. The model for emotion identification is trained on tweets from Wang et al. [10], but on a balanced distribution of emotion labels in order to counteract the issue of predicting underrepresented emotions in the training set. The size of the training set is 10,000 samples.

When presented with the system, the user can choose one of the multiple cities which are currently being considered in our work. Upon selecting, the user is presented with the overview screen. On the left, we list all local government tweets. This provides a quick look to what is the local government currently working on, ongoing projects and current issues. The user has the option to list all tweets

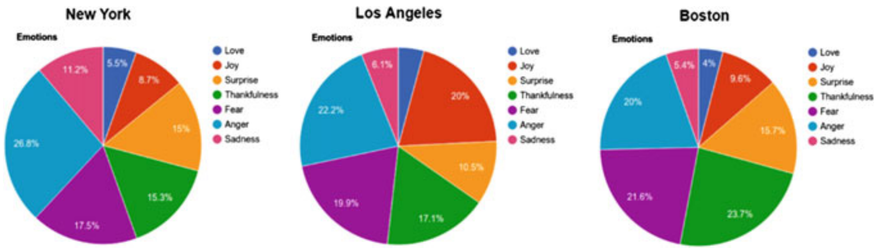


**Fig. 3** Emotion distribution in local government related tweets

mentioning any of the local government accounts or choose a specific tweet and take a look at all relevant messages to that specific tweet. These messages appear in the middle column. The system presents the content of each tweet, as well as the full and screen name of the user with the accompanying profile image. On the right part of the screen, a summary of the tweets in the middle column is given. We present an overview of the identified emotions in the relevant tweets which enables users to have a closer look at the local community's satisfaction with their city. If the user has chosen to list all tweets mentioning local government accounts, he is also presented with an emotional distribution by topic, in order to get even more meaningful insight. Topics are generated by applying hierarchical agglomerative clustering over the tweets. For this purpose, we utilize the *fastcluster*[3] library for hierarchical clustering. For scalability, we only consider words that have a frequency over a predefined threshold. We cut the dendrogram at a specific threshold value as well to extract the generated clusters.

In Fig. 3, the emotion distribution for New York, Los Angeles and Boston is presented for the tweets mentioning any of the corresponding local government accounts. We can observe that certain emotions are more dominant in some cities over others. For example, *joy* in San Francisco, *anger* in New York, *thankfulness* in Boston are more present in relation to the other cities. We leave more detailed analysis of the observed emotions for future work.

## 6 Conclusion

In this paper, we presented a convolutional neural network for emotion identification in Twitter messages and we applied it to monitoring emotions in tweets related to public local services. The model was evaluated on a set of hashtag labeled tweets with 7 distinct emotions. Using the presented architecture, we achieved improvements over current state-of-the-art performance with the dataset on reduced training set. Our approach obtains an accuracy of 50.12 and 55.77% with a training

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

set of 1000 and 10,000 samples. We trained the model on a set with a balanced distribution of emotion labels and we applied it to tweets related to public local services. We present a system that retrieves tweets from official local government accounts for several cities and related Twitter messages. We showcase how such system can be utilized by the local government and the general public to get insight into the current projects and problems and improve the communication between both parties.

**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).

## References

1. 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), 2015. IEEE, pp 52–57
2. Sobaci MZ, Karkin N (2013) The use of twitter by mayors in Turkey: tweets for better public services? Gov Inf Q 30(4):417–425 (Elsevier)
3. Agostino D (2013) Using social media to engage citizens: a study of Italian municipalities. Public Relat Rev 39(3):232–234 (Elsevier)
4. Flores CC, Rezende DA (2013) Strategic digital city: techno-social network twitter as communication channel for popular participation in city comprehensive plans. In: Proceedings of the nineteenth Americas conference on information systems, Chicago, Illinois
5. Kim Y (2014) Convolutional neural networks for sentence classification. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). Association for Computational Linguistics, pp 1746–1751
6. Pennington J, Socher R, Manning CD (2014) Glove: global vectors for word representation. In: Proceedings of the empirical methods in natural language processing (EMNLP 2014) 12:1532–1543
7. Roberts K, Roach MA, Johnson J, Guthrie J, Harabagiu SM (2012) EmpaTweet: annotating and detecting emotions on twitter. LREC 3806–3813
8. Balabantaray RC, Mohammad M, Sharma N (2012) Multi-class twitter emotion classification: a new approach. Int J Appl Inf Syst 4:48–53
9. 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
10. Wang W, Chen L, Thirunarayan K, Sheth AP (2012) Harnessing Twitter "big data" for automatic emotion identification. In: Privacy, security, risk and trust (PASSAT), 2012 international conference on and 2012 international conference on social computing (SocialCom). IEEE. 587–592
11. Sintsova V, Musat C-C, Pu FP (2013) Fine-grained emotion recognition in olympic tweets based on human computation. 4th Workshop on computational approaches to subjectivity, sentiment and social media analysis

12. Collobert R, Weston J, Bottou L, Karlen M, Kavukcuoglu K, Kuksa P (2011) Natural language processing (almost) from scratch. J Mach Learn Res (JMLR.org) 12:2493–2537
13. 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