Unsupervised Learning of Fundamental Emotional States via Word Embeddings

Conference Paper · December 2017

DOI: 10.1109/SSCI.2017.8280819

3 authors:

Mirko Mazzoleni
University of Bergamo

Gabriele Maroni
University of Bergamo

Fabio Previdi
University of Bergamo

Unsupervised Learning of Fundamental Emotional States via Word Embeddings

Mirko Mazzoleni*, Gabriele Maroni*, Fabio Previdi*
*University of Bergamo
Department of Management, Information and Production Engineering
Email: mirko.mazzoleni@unibg.it
Email: fabio.previdi@unibg.it

***Abstract*—This paper presents a novel approach for the detection of emotional states from textual data. The considered sentiments are those known as Ekman's basic emotions (Anger, Disgust, Sadness, Happiness, Fear, Surprise). The method is completely unsupervised and it is based on the concept of word embeddings. This technique permits to represent a single word through a vector, giving a methematical representation of the word's semantic. The focus of the work is to assign the percentage of the aforementioned emotions to short sentences. The method has been tested on a collection of Twitter messages and on the SemEval 2007 news headlines dataset. The entire period is expressed as the mean of the word's vectors that compose the phrase, after preprocessing steps. The sentence representation is finally compared with each emotion's word vector, to find the most representative with respect to the sentence's vector.**

I. Introduction

Writing has always been the medium through which humans entrusted their feelings to an everlasting support. Whether it is stone, paper, or, nowadays, the internet, the aim is always to leave trace of a man's passage. These footprints are today impressed for the major part on a social network canvas: the explosion of such platforms has indeed multiplied exponentially the available information in the form of binary (likes and clicks), image, video and, not least, textual data. It appeared immediately that such amount of data could be mined to extract useful information. In [1], the authors demonstrated how it is possible to predict personal traits from Facebook's likes, discussing also privacy issues. This knowledge can then be used for online content personalization and recommendation. If a like to a specific product can be directly associated with the person's interest, deriving this opinion or sentiment from unstructured data, such as text, can be difficult. However, efforts in this direction are well motivated: i) businesses always want to know public or consumer opinions about their products and services; ii) sentiments of tweets about politicians can be used to understand voters opinions [2]; iii) movie producers can mine Twitter data to know box-office movie revenues [3]. Traditionally, companies conduct consumer surveys for this purpose. Though well-designed surveys can provide quality estimations, they can be costly especially if a large volume of survey data is gathered. The ubiquity of social media services presents a great opportunity to understand the sentiment of the public, by analyzing its large-scale and opinion-rich data.

The methodologies which go under the name of "text mining" can be used to analyze textual data. The phrase "text mining" is generally used to denote any system that analyzes large quantities of natural language text and detects lexical or linguistic usage patterns [4]. Sentiment analysis or opinion mining is the computational study of people's opinions, appraisals, attitudes and emotions toward entities, individuals, events, topics and their attributes [5]. There has been extensive research on automatic text analysis for sentiment, such as sentiment classifiers [6], affect analysis [7], automatic survey analysis [8], opinion extraction [9]. These methods typically try to extract the overall sentiment revealed in a document, such as positive or negative. Apart from the dichotomic like/dislike sentiment case, a different approach consists of understanding the writer's emotion from his/her message. A basic set of universal emotions has been proposed by the psychologist Paul Ekman [10], who defined them from facial expressions. The Ekman's six fundamental emotions are Anger, Disgust, Sadness, Happiness, Fear and Surprise. Other scientist tried to define a finite set of emotional states. Carroll Izard [11] proposed to classify emotions as Anger, Contempt, Disgust, Distress, Fear, Guilt, Interest, Joy, Shame, and Surprise. Robert Plutchik [12] illustrated the primary ones as Anger, Anticipation, Disgust, Fear, Joy, Sadness, Surprise, and Trust.

Sentiment analysis can be performed with two different techniques: supervised and unsupervised algorithms [13]. Supervised methods train a classifier from labeled training data, while unsupervised methods does not require a "ground truth". In social media, it is time and labor consuming to obtain sentiment labels, whereas it is easy to collect vast quantities of unlabeled data. This makes unsupervised sentiment analysis essential for various applications. The employment of unsupervised methods for sentiment analysis of emotions has already been faced in [14]. Here the authors evaluated several different standard unsupervised techniques, such as Latent Semantic Analysis (LSA) [15] or Non-negative Matrix Factorization (NMF) [16]) to tag each sentence with four emotions (Anger, Fear, Joy, Sadness) that are common to all the used datasets.

Recently, word embeddings techniques emerged as a new unsupervised learning paradigm for Natural Language Processing. With these methods, words or phrases from the vocabulary are mapped to vectors of real numbers, capturing semantic similarities between them. Examples are the Word2Vec [17] and Glove [18] models. These vector-based approaches have been used to perform sentiment analysis, but *only* to discern positive vs. negative sentiment [19].

It follows naturally that the next step is to combine the aforementioned approaches. The innovative contribution of this paper is to present a completely unsupervised procedure to perform *emotion detection* of short sentences *based on word embeddings*. The method has been tested on twitter data, collected given a specified keyword and annotated manually with one of the six emotions, by 11 annotators. The method has been tested also on the SemEval 2007 news headline dataset, to provide a comparison with previous research. After a pre-processing step, every word of each tweet has been trasformed into a vector. The vector of the entire sentence has been computed as the mean of the word's vector that composed the sentence. The obtained sentence's vector is then compared by cosine similarity with the vector of each Ekman's basic emotion. The emotion which gave the highest similarity score with the sentence vector was chosen as the representative emotion of the tweet. Results show how the method is able to identify the correct sentiment in a dichotomous setting, and comparable to existing methods on a more challenging problem.

The remainder of the paper is organized as follows. Section II formally presents the problem and introduces the main notations. Section III discusses briefly the word embeddings concept adopted for this work. In Section IV, the developed algorithm rationale is described. Section V highlights the main results of the proposed approach, evaluated on a set of manually labeled tweets and on the SemEval 2007 news headlines dataset. Lastly, Section VI is devoted to concluding remarks and future developments.

II. Problem statement

The purpose of this work is to detect emotions in short sentences such as tweet messages and news headlines. The chosen emotions are the six Ekman's basic ones. It is possible to obtain a threefold output. The first one is a "soft" information, consisting in the percentage of each emotion contained in the sentence. The second output is a "hard" information, represented by tagging a message with only the dominant emotion. The third output is a binary one, given by the percentage of positive and negative contents in the message. Positive content is given by the Happiness and Surprise emotions. Negative content is given by Anger, Disgust, Sadness and Fear categories. The proposed method is based on the concept of word embeddings. This permits to represent each string word w in the message as a d-dimensional vector. The word w_j of the sentence t is represented with the vector $\mathbf{v}_{j,t} \in \mathbb{R}^{d\times 1}$. The vector representing the entire sentence $\mathbf{x}_t \in \mathbb{R}^{d\times 1}$ is computed as the sum of the vectors of each sentence's word, $\mathbf{x}_t = \sum_{j=1}^{n_t} \mathbf{v}_{j,t}$, with n_t the length (number of words) of the sentence t. This operation is motivated by the fact that the linearity of the vector operations seems to weakly hold also for the addition of several vectors, so it is possible to add several word or phrase vectors to form representation of short sentences [17]. The sentence's vectors $\mathbf{x}_t$ are collected in the matrix $X = [\mathbf{x}_1\ \mathbf{x}_2\ \dots\ \mathbf{x}_N]^T \in \mathbb{R}^{N\times d}$, with N the total number of messages. Each considered emotion is expressed by a word in natural language. Therefore, it makes sense to represent the emotion i as its corresponding word embedding $\mathbf{e}_i \in \mathbb{R}^{d\times 1}$. The emotions' vectors are collected in the matrix $E = [\mathbf{e}_1\ \mathbf{e}_2\ \dots\ \mathbf{e}_M]^T \in \mathbb{R}^{M\times d}$, where M is the number of emotions. For each message t, the corresponding percentage of each emotion is computed, forming the vector $\mathbf{s}_t \in \mathbb{R}^{M\times 1}$. The computed emotions' percentages for each sentence are collected into the matrix $S = [\mathbf{s}_1\ \mathbf{s}_2\ \dots\ \mathbf{s}_N]^T \in \mathbb{R}^{N\times M}$.

The presented work investigates the connection between lexical semantics and emotions. While some words have emotional meaning with respect to an individual story, for many others the affective power is part of the collective imagination (e.g. words such as "mum", "ghost", "war").

III. Word embeddings

A word embedding $W : w \to \mathbb{R}^d$ is a paramaterized function mapping words w to vectors in $\mathbb{R}^d$. The function is typically a lookup table parameterized by a matrix $U \in \mathbb{R}^{D\times d}$. Each row of U is one of the D words in the dictionary or corpus on which the algorithm has been trained. Word embeddings were first introduced in [20]. The authors in [17], [21] have contributed to their widespread use by proposing an efficient algorithm, named Word2Vec, to learn word vectors. The vectors are learned in such a way that a similar representation is attributed to words that appear in the same context. Context is the set of words that appear before and after the considered one. Actually, Word2Vec proposes two different learning architectures. Both of them are neural networks models. The first architecture is the Continuous Bag-Of-Words (CBOW) model, as depicted in Figure 1. The aim of this representation is to predict the current word w_j given its context $C(w_j)$.

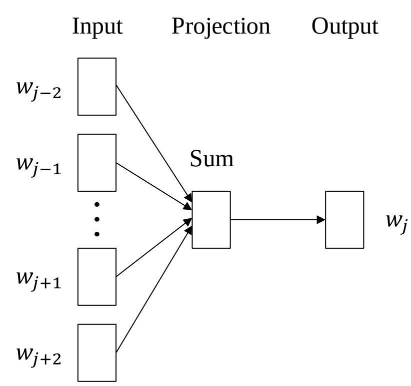

Fig. 1: The CBOW architecture predicts the current word w_j based on the context $C(w_j)$

The second architecture is the Skip-gram model, see Figure 2. This representation tries to predict the context words $C(w_j)$ given the current word w_j. Recommended dimensions for the context are 10 word for the CBOW model and 5

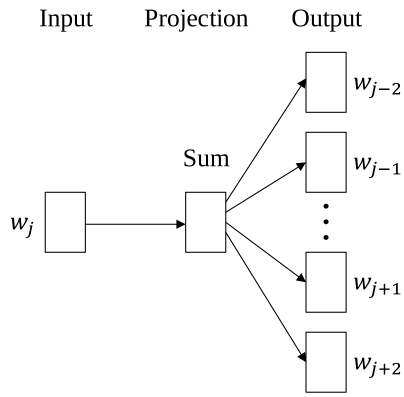

Fig. 2: The Skip-gram model architecture learns word vector representations that are good at predicting the nearby words

words for the Skip-gram one [21]. Word embeddings exhibit a remarkable property: analogies between words seem to be encoded in the difference vectors between words. For example, there seems to be a constant male-female difference vector:

$$W(\text{woman}) - W(\text{man}) \approx W(\text{aunt}) - W(\text{uncle})$$
$$W(\text{woman}) - W(\text{man}) \approx W(\text{queen}) - W(\text{king})$$

Word embeddings are therefore able to encode semantic relationships between words. Words that appear in the same context will be closer in the vector space. The idea is then to consider the closeness of a sentence's vector to an emotion's vector as indicator of the content degree of that emotion in a particular message.

IV. Emotion detection

For the application presented in this work, we used pre-trained word vectors: the employed embedding is the one delivered by Google, trained on a GoogleNews corpus with the CBOW architecture.[1]The word vectors have a dimension $d = 300$. The Glove vector embeddings were also tested with lower results. The corpus has been considered general enough to provide unbiased word's context representation.
The output of the unsupervised algorithm is the sentiment matrix $S \in \mathbb{R}^{N\times M}$, where $M = 6$ Ekman's emotions and N depends on the considered dataset. The application for the emotion detection task has been developed in the Python 2.7 programming language. The Gensim Python library was used for the management of word embeddings. The preliminar steps have the aim to preprocess and clean the raw text data. The result is a cleaned sentence t from a raw sentence $\tilde{t}$. The entire process is subdivided in the following steps.

A. Tokenization

The first action is the tokenization of the raw sentence $\tilde{t}$. This permits to extract the single words that compose a message. The isolation of the single words happens at a blank space. For the tweets dataset, the tokenizer took care of deleting the handles (@) used for referencing to users. We chose to not perform lowercase conversion of text, since capital letters can convey emotions such as anger.

B. Emoticon replacement

The emoticons used in text messages have an high emotional content. For this reason, emoticon were grouped into six categories, one for each emotion. Then, they were substituted with the word associated with the representative emotion.

C. Stopwords removal

Common stopwords were removed from the text. The removed common english words are those present in the Python library NLTK. As advocated in [19], we did not remove negative stopwords such as "not" and "no" since they are indicative of sentiment. Question ("?") and exclamation mark ("!") , along with suspension dots ("…") were not removed for the same reason.

D. Non-english words removal

Non-english words were removed from the text since they are not present in the used word embedding dictionary, and so they do not provide any information. The dictionary used to search for english word is the one provided with the Python library PyEnchant. Single letters and numbers were also removed.

E. Stemming

Stemming was applied only for english words that were not present in the word embedding dictionary. In this case, it is not possible to associate a vector to this word. A solution to this problem can be to stem the word and check if the stemmed word is present in the embeddings. Otherwise, the word is discarded.

F. Sentiment computation

After the steps IV-A – IV-E, the cleaned sentence t is formed. The sentence is composed by all the remaining w_j words, $j = 1 \dots n_t$. The representative vector of the each sentence t, $\mathbf{x}_t \in \mathbb{R}^{d\times 1}$, is computed as described in Section II. By using the sentences matrix X and the emotions matrix E, the cosine similarity between each sentence and each emotion is computed. The results are collected in the vector $\mathbf{y}_t \in \mathbb{R}^{M\times 1}$, where $y_{m,t}$ is the cosine similarity of emotion m with the sentence t, $m = 1, \dots, M$. Since we want to express the percentage content of each emotion given a specific sentence, the obtained results have to be normalized to sum to one. This can be obtained with the following relations:

$$\left\| \frac{\mathbf{y}_t}{\|\mathbf{y}_t\|} \right\|^2 = 1^2 \Rightarrow \left\| \frac{\mathbf{y}_t}{\sqrt{\sum_{m=1}^{M} y_{m,t}^2}} \right\|^2 = \sum_{j=1}^{M} \frac{y_{m,t}^2}{\sum_{m=1}^{M} y_{m,t}^2} = 1$$

The M emotions' percentages $s_{m,t} = \frac{y_{m,t}^2}{\sum_{m=1}^{M} y_{m,t}^2}$ for the sentence t are then collected in the vector $\mathbf{s}_t \in \mathbb{R}^{M\times 1}$.

[1]https://code.google.com/archive/p/word2vec/ - Last accessed: 19 Jun 2017

V. Results and discussion

A. Twitter dataset

A web-application was developed in order to collect Twitter messages. The tweets were downloaded by specifying a keyword. In this experiment, the keyoword was the word "Christmas". A total of 70 tweets were manually labeled by 11 annotators. Each annotator tagged each tweet with one of the six emotions. The majority vote was taken as the emotion label for that tweet. In case of multiple candidate labels, the tweet was discarded. The final dataset consisted in 64 tweets. For the six emotions, the following words were chosen as representatives: 1) Anger: "anger", 2) Disgust: "disgust", 3) Fear: "fear", 4) Happiness: "happiness", 5) Sadness: "sadness", 6) Surprise: "wonderment". After the computation of the emotional content for each tweet, the emotion with the highest probability is taken as category for that tweet. A multiclass classification report is shown in Table I. The report highlights the precision, recall and F1 indicators. The average results are weighted by the number of the elements in each class. The messages have been labeled for the main part with the Happiness category (43 out of 64). No tweets were labeled with the Fear emotion.

TABLE I: Twitter dataset multiclass results

MULTICLASS				
Emotion	Prec.	Rec.	F1	Number
ANGER	0.00	0.00	0.00	5
DISGUST	100.00	20.00	33.00	5
FEAR	0.00	0.00	0.00	0
HAPPINESS	76.00	65.00	70.00	43
SADNESS	33.00	40.00	36.00	5
SURPRISE	22.00	33.00	27.00	6
AVERAGE/TOTAL	**63.00**	**52.00**	**55.00**	**64**

Another result that it is possible to exploit is the dichotomic classification case. In this setting, only two classes are present: positive and negative sentiment. In the positive class are grouped the tweets labeled with the category Happiness and Surprise. The Anger, Disgust, Fear and Sadness categories constitute the negative class. Table II reports the binary classification results.

TABLE II: Twitter dataset binary results

BINARY CLASS				
Emotion	Prec.	Rec.	F1	Number
NEGATIVE	50.00	60.00	55.00	15
POSITIVE	87.00	82.00	84.00	49
AVERAGE/TOTAL	**78.00**	**77.00**	**77.00**	**64**

TABLE III: SemEval 2007 Affective task results [22]

	Fine r	Prec.	Rec.	Coarse F1
ANGER				
WN-AFFECT PRESENCE	12.08	**33.33**	3.33	6.06
LSA SINGLE WORD	8.32	6.28	63.33	11.43
LSA EMOTION SYNSET	17.80	7.29	86.67	13.45
LSA ALL EMOTION WORDS	5.77	6.20	**88.33**	11.58
NB TRAINED ON BLOGS	19.78	13.68	21.67	**16.77**
SWAT	24.51	12.00	5.00	7.06
UA	23.20	12.74	21.6	16.03
UPAR7	32.33	16.67	1.66	3.02
EMBEDDINGS	**37.68**	0.00	0.00	**-**
DISGUST				
WN-AFFECT PRESENCE	-1.59	0.00	0.00	-
LSA SINGLE WORD	13.54	**2.41**	70.59	**4.68**
LSA EMOTION SYNSET	7.41	1.53	64.71	3.00
LSA ALL EMOTION WORDS	8.25	1.98	**94.12**	3.87
NB TRAINED ON BLOGS	4.77	0.00	0.00	-
SWAT	18.55	0.00	0.00	-
UA	16.21	0.00	0.00	-
UPAR7	12.85	0.00	0.00	-
EMBEDDINGS	**29.35**	0.00	0.00	**-**
FEAR				
WN-AFFECT PRESENCE	24.86	**100.00**	1.69	3.33
LSA SINGLE WORD	29.56	12.93	**96.61**	22.80
LSA EMOTION SYNSET	18.11	12.44	94.92	22.00
LSA ALL EMOTION WORDS	10.28	12.55	86.44	21.91
NB TRAINED ON BLOGS	7.41	16.67	3.39	5.63
SWAT	32.52	25.00	14.40	18.27
UA	23.15	16.23	26.27	20.06
UPAR7	**44.92**	33.33	2.54	4.72
EMBEDDINGS	37.45	32.00	33.00	**32.00**
JOY				
WN-AFFECT PRESENCE	10.32	**50.00**	0.56	1.10
LSA SINGLE WORD	4.92	17.81	47.22	25.88
LSA EMOTION SYNSET	6.34	19.37	72.22	30.55
LSA ALL EMOTION WORDS	7.00	18.60	**90.00**	30.83
NB TRAINED ON BLOGS	13.81	22.71	59.44	**32.87**
SWAT	26.11	35.41	9.44	14.91
UA	2.35	40.00	2.22	4.2
UPAR7	22.49	54.54	6.66	11.87
EMBEDDINGS	**40.33**	40.00	4.00	7.00
SADNESS				
WN-AFFECT PRESENCE	8.56	33.33	3.67	6.61
LSA SINGLE WORD	8.13	13.13	55.05	21.20
LSA EMOTION SYNSET	13.27	14.35	58.71	23.06
LSA ALL EMOTION WORDS	10.71	11.69	**87.16**	20.61
NB TRAINED ON BLOGS	16.01	20.87	22.02	21.43
SWAT	38.98	32.50	11.92	17.44
UA	12.28	25.00	0.91	1.76
UPAR7	**40.98**	**48.97**	22.02	**30.38**
EMBEDDINGS	23.92	0.00	0.00	-
SURPRISE				
WN-AFFECT PRESENCE	3.06	13.04	4.68	6.90
LSA SINGLE WORD	9.71	6.73	67.19	12.23
LSA EMOTION SYNSET	12.07	7.23	89.06	13.38
LSA ALL EMOTION WORDS	12.35	7.62	**95.31**	14.10
NB TRAINED ON BLOGS	3.08	8.33	1.56	2.63
SWAT	11.82	11.86	10.93	11.78
UA	7.75	**13.70**	16.56	**15.00**
UPAR7	**16.71**	12.12	1.25	2.27
EMBEDDINGS	12.25	9.00	5.00	6.00

B. SemEval 2007 dataset

The SemEval 2007 task on "Affective text" focused on the emotion classification of news headlines extracted from news web sites [23]. Headlines are suitable for these experiments because they are typically intended to express emotions, in order to draw the readers' attention. The dataset is composed by 250 training news and 1000 test news. Each news is labeled by six annotators with a percentage for each emotion. The performance is assessed through fine-grained and coarse-grained metrics. Fine-grained evaluations were conducted using the Pearson measure of correlation between the system scores and the gold standard scores, for each emotion. For the coarse-grained evaluations, where each emotion was mapped to a $0/1$ classification ($0 = [0, 50), 1 = [50, 100]$), the precision, recall, and F-measure metrics are computed. This is translated to a multilabel classification problem, where each sentence can have more than one category assigned. Table III reports the results, for each emotion, of the proposed method compared to previous researches [22], providing both fine-grained and coarse-grained evaluations. In accordance with the authors in [22], we used the word "joy" to represent the emotion Happiness. For the six emotions, the following words were chosen as representatives: 1) Anger: "anger", 2) Disgust: "disgust", 3) Fear: "fear", 4) Happiness: "joy", 5) Sadness: "sadness", 6) Surprise: "wonderment". Table III reports a comparison of the results for each of the defined emotion categories. Table IV shows an aggregate overall results of the different methods, averaging other all the emotions. The average method is a simple (non-weighted) arithmetic mean.

TABLE IV: SemEval 2007 Affective task results - Overall average

	Fine			Coarse
	r	Prec.	Rec.	F1
WN-AFFECT PRESENCE	9.54	**38.28**	1.54	4.00
LSA SINGLE WORD	12.36	9.88	66.72	16.37
LSA EMOTION SYNSET	12.50	9.20	77.71	13.38
LSA ALL EMOTION WORDS	9.06	9.77	**90.22**	**17.57**
NB TRAINED ON BLOGS	10.81	12.04	18.01	13.22
SWAT	25.41	19.46	8.61	11.57
UA	14.15	17.94	11.26	9.51
UPAR7	28.38	27.60	5.68	8.71
EMBEDDINGS	**30.17**	13.50	6.82	7.50

The different methods reported in [22] are:

1) WN-AFFECT PRESENCE, which annotates the emotions in a text simply based on the presence of words from the WordNet Affect lexicon [24]
2) LSA SINGLE WORD, which calculates the LSA similarity between the given text and each emotion, where an emotion is represented as the vector of the specific word denoting the emotion (e.g., "joy")
3) LSA EMOTION SYNSET, where in addition to the word denoting an emotion, its synonyms from the WordNet synset are also used
4) LSA ALL EMOTION WORDS, which augments the previous set by adding the words in all the synsets labeled with a given emotion, as found in WordNet Affect
5) NB TRAINED ON BLOGS, which is a Naive Bayes classifier trained on the blog data annotated for emotions
6) UPAR7 [25] is a rule-based system using a linguistic approach
7) UA [26] uses statistics gathered from three search engines (MyWay, AlltheWeb and Yahoo) to determine the kind and the amount of emotion in each headline
8) SWAT [27] is a supervised system using an unigram model trained to annotate emotional content

C. Discussion

Authors are aware that the emotions identified by Ekman are related to facial expressions. There is no guarantee that they are portable to the sentiment of textual data. However, their use in sentiment analysis as been advocated in previous research [28], [14], [29].

The results of the multiclass problem on the Twitter dataset, see Table I, show that the method can discern well the different six categories. When faced with the binary classification task, see Table II, the proposed method reaches considerable results, despite the classes are quite unbalanced.

Regarding the SemEval 2007 Affective task, as expected, different systems have different strengths. In terms of performance for individual emotions, reported in Table III, the system based on blogs gives the best results for Joy, which correlates with the size of the training data set (Joy had the largest number of blogposts). The blogs are also providing the best results for anger (which also had a relatively large number of blogposts). This system is however a supervised one. Our model is the best by a fair amount for the Fear emotion, and is the best in the fine-grained performance for 3 out of 6 emotions. By looking at the aggregated results in Table IV, the system based exclusively on the presence of words from the WordNet Affect lexicon has the highest precision at the cost of low recall. Instead, the LSA system using all the emotion words has by far the largest recall and is the overall best coarse-grained performer, although the precision is significantly lower. Our systems give the best performance in terms of fine-grained evaluation, perhaps due to the deep semantic analysis performed by the word embeddings.

The overall comparison of the two datasets and different evaluation task shows that:

1) The proposed method is good when faced with multi-class classification problem
2) The proposed method performs very well with the fine-grained evaluation, that is, suggesting the percentage of one emotion in the sentence relative to the other ones
3) The proposed method does not perform well on the multilabel classification problem

VI. Conclusions

This paper presented an approach based on word embeddings for the detection of emotions in text. Word embeddings are a recent and promising Natural Language Processing method based on neural networks to represent a word given its context. This model is able to incorporate rich semantic relationship which can be useful for sentiment analysis. The proposed method, after a cleaning process, represented each word in the sentence as a vector. Then, the sentence is represented as the sum of the words' vectors. The considered emotions are represented by the vector embeddings of associated words. The cosine similarity is computed for each pair of sentence and emotion vectors, to find the relative emotional content. The result is then standardized to trasform each score into a positive number and to sum to one. The final result is then considered as the probability of each emotion to be present in the sentence. The methodology is tested on two different datasets. The first dataset consisted in a collection of tweets crawled by means of a developed application. The tweets were labeled by 11 annotators with the prevalent emotion. On this dataset, the algorithm was tested in a binary and multiclass classification setting. The second dataset consisted in the SemEval 2007 Affective task, which contains a set of headline news. In this case, the method was tested as defined by the proposers of the competition, to make possible a comparison with past methods. The fine-grained evaluation computed the correlation between the predicted and annotated emotions' percentages. The coarse-grained evaluation tested the models in a multilabel classification problem. Results show how the method based on word embeddings performs well on the multiclass and correlation tasks, while exhibits a poor performance on the multilabel classification problem. Further research is devoted to learning word embeddings more representative for the sentiment task, and the incorporation with other techniques.

References

[1] M. Kosinski, D. Stillwell, and T. Graepel, "Private traits and attributes are predictable from digital records of human behavior," *Proceedings of the National Academy of Sciences*, vol. 110, no. 15, pp. 5802–5805, 2013.

[2] B. O'Connor, R. Balasubramanyan, B. R. Routledge, and N. A. Smith, "From tweets to polls: Linking text sentiment to public opinion time series." *ICWSM*, vol. 11, no. 122-129, pp. 1–2, 2010.

[3] S. Asur and B. A. Huberman, "Predicting the future with social media," in *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010 IEEE/WIC/ACM International Conference on*, vol. 1. IEEE, 2010, pp. 492–499.

[4] F. Sebastiani, "Machine learning in automated text categorization," *ACM computing surveys (CSUR)*, vol. 34, no. 1, pp. 1–47, 2002.

[5] B. Liu and L. Zhang, "A survey of opinion mining and sentiment analysis," in *Mining text data*. Springer, 2012, pp. 415–463.

[6] S. Das and M. Chen, "Yahoo! for amazon: Extracting market sentiment from stock message boards," in *Proceedings of the Asia Pacific finance association annual conference (APFA)*, vol. 35. Bangkok, Thailand, 2001, p. 43.

[7] P. Subasic and A. Huettner, "Affect analysis of text using fuzzy semantic typing," *IEEE Transactions on Fuzzy systems*, vol. 9, no. 4, pp. 483–496, 2001.

[8] H. Li and K. Yamanishi, "Mining from open answers in questionnaire data," in *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2001, pp. 443–449.

[9] S. Morinaga, K. Yamanishi, K. Tateishi, and T. Fukushima, "Mining product reputations on the web," in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2002, pp. 341–349.

[10] P. Ekman, "Facial expression and emotion." *American psychologist*, vol. 48, no. 4, p. 384, 1993.

[11] G. J. Boyle, "Reliability and validity of izard's differential emotions scale," *Personality and Individual Differences*, vol. 5, no. 6, pp. 747–750, 1984.

[12] R. E. Plutchik and H. R. Conte, *Circumplex models of personality and emotions.* American Psychological Association, 1997.

[13] J. Friedman, T. Hastie, and R. Tibshirani, *The elements of statistical learning*. Springer series in statistics Springer, Berlin, 2001, vol. 1.

[14] S. M. Kim, A. Valitutti, and R. A. Calvo, "Evaluation of unsupervised emotion models to textual affect recognition," in *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*. Association for Computational Linguistics, 2010, pp. 62–70.

[15] T. K. Landauer, *Latent semantic analysis*. Wiley Online Library, 2006.

[16] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Advances in neural information processing systems*, 2001, pp. 556–562.

[17] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[18] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation." in *EMNLP*, vol. 14, 2014, pp. 1532–1543.

[19] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*. Association for Computational Linguistics, 2011, pp. 142–150.

[20] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A neural probabilistic language model," *Journal of machine learning research*, vol. 3, no. Feb, pp. 1137–1155, 2003.

[21] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[22] C. Strapparava and Mihalcea, "Learning to identify emotions in text," in *Proceedings of the 2008 ACM symposium on Applied computing*. ACM, 2008, pp. 1556–1560.

[23] C. Strapparava and R. Mihalcea, "Semeval-2007 task 14: Affective text," in *Proceedings of the 4th International Workshop on Semantic Evaluations*. Association for Computational Linguistics, 2007, pp. 70–74.

[24] C. Strapparava, A. Valitutti, *et al.*, "Wordnet affect: an affective extension of wordnet." in *LREC*, vol. 4, 2004, pp. 1083–1086.

[25] F.-R. Chaumartin, "Upar7: A knowledge-based system for headline sentiment tagging," in *Proceedings of the 4th International Workshop on Semantic Evaluations*. Association for Computational Linguistics, 2007, pp. 422–425.

[26] Z. Kozareva, B. Navarro, S. Vázquez, and A. Montoyo, "Ua-zbsa: a headline emotion classification through web information," in *Proceedings of the 4th international workshop on semantic evaluations*. Association for Computational Linguistics, 2007, pp. 334–337.

[27] P. Katz, M. Singleton, and R. Wicentowski, "Swat-mp: the semeval-2007 systems for task 5 and task 14," in *Proceedings of the 4th international workshop on semantic evaluations*. Association for Computational Linguistics, 2007, pp. 308–313.

[28] F. Keshtkar and D. Inkpen, "A corpus-based method for extracting paraphrases of emotion terms," in *Proceedings of the NAACL HLT 2010 Workshop on Computational approaches to Analysis and Generation of emotion in Text*. Association for Computational Linguistics, 2010, pp. 35–44.

[29] S. M. Mohammad and P. D. Turney, "Emotions evoked by common words and phrases: Using mechanical turk to create an emotion lexicon," in *Proceedings of the NAACL HLT 2010 workshop on computational approaches to analysis and generation of emotion in text*. Association for Computational Linguistics, 2010, pp. 26–34.