Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 83 (2016) 292 – 299

Procedia Computer Science

The 7th International Conference on Ambient Systems, Networks and Technologies (ANT 2016)

Using YouTube comments for text-based emotion recognition

Douiji yasmina[a]*, Mousannif Hajar[b], Al Moatassime Hassan[a]

[a]*Faculty of Science and Technology, Abdelkarim Elkhattabi Street, Guéliz, Marrakesh P.C 40549, Morocco*
[b]*Faculty of Semlalia, Prince My Abdellah Street,Marrakesh P.C 42390, Morocco*

Abstract

With the increase of Smartphone use, there is a growing need for advanced features that offer to Smartphone users a smarter interaction. We aim through the presented system to detect users' emotions from their textual exchanges, dealing with the complexity of chat writing style and the evolution of languages. We consider that such a system is a start for interesting applications that exploit users' emotional states. Our system uses an unsupervised machine learning algorithm that performs emotion classification, based on a data corpus built from YouTube comments. The reason behind such a choice is the similarity between YouTube comments and instant messages writing style. To classify a text entry into a particular emotion category, we compute its similarity to each target emotion, using the Pointwise Mutual Information measure. Our method yields a global precision of 92.75%, which reflects the feasibility of our approach.

Peer-review under responsibility of the Conference Program Chairs

Keywords: affective computing; emotion recognition; text; machine learning.

1. Introduction

YouTube comments present a rich resource for publicly available text. They hold different styles of expression, and are in almost all existing languages. They also raise different issues: opinions, stories and Emotions. The empirical research we present in this paper harnesses the huge potential in YouTube comments for text-based emotion detection. Former experiences have shown that it is quite complicated when it comes to emotion extraction

* Corresponding author. Tel.: +212648714788; fax: N/A.
E-mail address: yasmina.douiji@ced.uca.ma

Peer-review under responsibility of the Conference Program Chairs
doi:10.1016/j.procs.2016.04.128

from text as compared to emotion detection using face[1], voice[2] and gestures[3]. Among the difficulties that hinder text-based emotion detection, there are: the complexity of natural language, its continuous evolution (new expressions everyday), and the ambiguous context of the author.

In this paper, we use an unsupervised machine learning algorithm based on the previous work of[4], to which we brought modifications we later discuss in this paper. We classify emotions according to the six basic emotions of Ekman[5]. Each emotion category is represented by a list of expressive words. And to determine the emotion expressed in a piece of text, we first classify its component words. We start by extracting Adjectives, Nouns, Verbs and Adverbs (NAVA words) from text. The other words (pronouns, interjections, prepositions…) are not considered, because they are all the time neutral. We, then, compute the probabilities for each word to belong to each emotion category. The probability of a single word to belong to a particular emotion category is the value of the normalized form of the PMI (Pointwise Mutual Information), between this word and the representative words of the emotion category. In fact, unlike the previous work[4] we prefer to use the normalized form because it gives better insights about the relatedness between two events[6]. When a negation is present, the concerned word is automatically assigned to the category "Neutral" (probability equal to zero). The probabilities of the whole sentence are the average of the obtained probabilities by the number of classified words.

The data corpus that we use to compute the different PMIs is built by importing comments from YouTube using YouTube API version 3. To ensure having enough rich content in the corpus, we browse videos from different YouTube categories (divertissement, Blogs & People …) using keywords relevant to the six emotions of Ekman. Once videos identifiers are retrieved, we import the corresponding comments.

Our system shows satisfying results compared to previous works. First conducted tests give high precision ranging from 91% to 95% for different target emotions. To run tests, we choose two different types of sentences. The first type contains affective words that correspond to each of the target emotions, and the second type does not contain any affective words.

The remainder of this paper is organized as follows: the second section presents previous works in text-based emotion detection, giving their results and accuracy issues. The third section explains, in details, the different steps of our approach, including the process of building the data corpus, the algorithm used and tests organization. Results of our study, as well as, discussions are presented in the fourth section. And finally, we give outlines for possible improvements in the last section.

2. Previous work

Different methods were used to build emotion detection systems from text, which can be grouped into three categories of algorithms[7]: Knowledge-based, machine learning based, and hybrid. In this section, we adopt the same classification scheme, but we present works that are more or less related to the work we develop in this paper.

2.1. Knowledge-based detection

This category of methods consist of using affect lexicons, such as WordNet-affect[8] or General Inquirer[9], and a baseline algorithm in order to check the presence of affective words, present in the lexicon, in the text to classify. The algorithm, then, computes a score that reflects frequency in the text for each detected word. To detect the emotion expressed in a piece of text (e.g. in a sentence), words' scores are aggregated following some linguistic rules: Authors in[10] for example, perform sentiment tagging on news headlines. They consider that the root word, which does not depend on any other word in the headline, has the most important contribution on the global subject of the headline. So they extract the root word using Stanford Dependency Parser[11], and multiply its emotional score by 6. They also search patterns like (noun → subject → verb) and (verb → direct object → noun) in the dependency graph with verbs that increase or decrease a quantity, in order to increase or decrease the score of depending nouns. The presented approach constitutes a viable method for emotion detection, the only concern about such approach is that the rules are not comprehensive and need to be manually defined.

2.2. Machine learning based methods

Texts are classified into a particular emotion category, using either supervised or unsupervised machine learning algorithms:

2.2.1. Supervised machine learning algorithms

The most used supervised machine learning algorithms in the context of text-based emotion detection is SVM: Authors in[12] use the SMO implementation of SVM provided by WEKA Software for emotion detection from new text entries. As training dataset, they build a heterogonous dataset containing news headlines, fairy tales and blogs. In addition to SMO, they run other algorithms available in the same software, such as, J48 for Decision Trees and Naïve Bayes for the Bayesian classifier. The result of their experiments shows that SMO performs statistically better than the other algorithms with a confidence level of 95% based on the accuracy rate. Authors in[13] end up with the same result comparing SVM and baseline algorithm. Their system performs an accuracy of 73.89% using manually annotated blog posts as training dataset. However SVM is not the only supervised machine learning algorithm that can perform well as shown in[14]. In this study, authors use, in addition to SVM, the Conditional Random Field (CRF) algorithm[15]. As feature set for both algorithms, they use keywords from an affective lexicon. And in the case of CRF, they add as feature the emotion being expressed by the previous sentence in the blog post, which reflects the context of the sentence to classify. The result of this study shows that CRF outperforms SVM, meaning that the emotions expressed in nearby sentences affect each other.

Supervised methods acquire a massive manual participation, especially to annotate sentences in the training dataset, which leads to another problem that is the agreement between judges that annotate sentences that never reaches 100%.

2.2.2. Unsupervised machine learning algorithms

Consist of using statistical measures in order to compute the semantic relatedness between words of a given sentence and the target emotions. Our empirical study is based on the work of[4], which use an unsupervised method that considers semantic and syntactic relations to detect emotions. This method does not require a pre-trained dataset. It consists of measuring the Pointwise Mutual Information parameter (PMI) between each word in the text to classify and representative words of each target emotion. This measurement is based on the co-occurrence between the word to classify and the representative words in the corpus.

2.3. Hybrid methods

Hybrid methods combine both knowledge-based and machine learning methods, as shown in[16]. Authors in this empirical study use as feature set for their learning module: semantics related to specific emotions, instead of simple keywords, which are extracted from the text to classify thanks to a rule-based approach. This method outperforms previous approaches. The target emotions are, however, limited.

As described in this section, there are three different methods to detect emotions from text, and each of them has certain limitations. In general, we notice that the aforementioned systems are limited to the six emotions of Ekman and do not suggest a way to adapt their systems to bigger range of emotions or to other emotion representations. Moreover, they do not consider the issue of language evolution (appearance of new expressions). In our work, we try to deal with the complexity of spoken language but also its continuous evolution, to end up with most accurate emotion inference.

3. Methodology

Our approach consists of two main phases: first, we start by building a data corpus to train our system using YouTube comments. Emotion labeling is not required in this phase; instead we manage to build a heterogeneous affect-corpus that contains different kinds of emotion expressions, by submitting a set of pre-defined requests to

YouTube API. This phase is very important as it influences the performance of the used algorithm. The second phase consists of running our unsupervised machine learning algorithm to classify new text entries. We give more details about these phases in the sub-sections below:

3.1. Corpus building

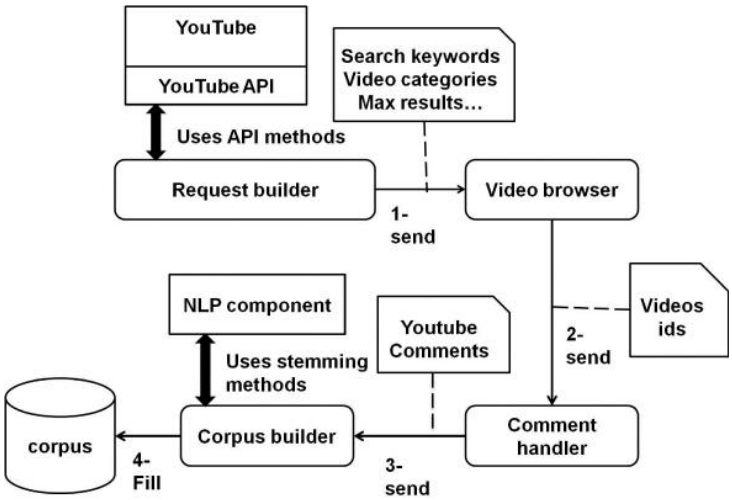

Fig. 1. Building the data corpus in four steps.

To train our classifier, we suggest YouTube comments. We assume that YouTube comments are a rich resource of natural expressions of feelings, thoughts and opinions. They are also close to writing style of instant messages or SMS, which are the objective of our application[17, 18]. In this empirical study, we used YouTube API v3, which allows submitting different kinds of requests, such us: browsing videos using keywords, retrieving videos IDs, descriptions, comments, etc.

In order to end up with enough diversified content, we build many requests through the "Request Builder" component. In each request, we specify a different combination of keywords, and video categories such as: Entertainment", "Movies & Animation", "News & Politics", etc. For example, in order to obtain comments that are relevant to the emotion "sadness", we use as keywords: "sad or heartbreaking" and so on. The requests thus built, are sent to the "video browser" component, which is going to retrieve videos Ids. Once retrieved, videos Ids are sent to the "Comment handler" component that extracts comments based on those Ids. Finally, the obtained comments are stored in our data corpus thanks to the "corpus builder" component. The role of the last component is to perform data words stemming on the comments using methods from the "NLP Component". Stemming gives the same form for words from the same family, for example: "am", "are" and "is" become "be", plural nouns become singular, present continuous forms of verbs become infinitive forms and so on. We run many tests over sentences expressing a particular emotion and we, finally, compare the obtained results and choose the best request parameters (keywords and video categories Ids) that generate most relevant content to the target emotion. For example, we notice that the categories "education" and "Blogs & People" with appropriate keywords give us most relevant content for the emotion category "disgust".

3.2. Text-based emotion detection algorithm

3.2.1. Emotion classification at word level

To classify a piece of text, we first perform a word level classification. The algorithm we use in this study is based on the previous work of[4]. It is an unsupervised machine learning method that computes the relatedness

between the word to classify *(w)* and a particular emotion *(e_j)*, using the Pointwise Mutual Information parameter (PMI). The basics of the algorithm we propose are as follows:

- For each emotion category *(e_j)*, we consider a set of representative words *(r_{ij})*. For example, fear is represented by the words ("fright", "terror", "scare", "fear").
- Let us consider S_w, the set of sentences in our corpus that contain at least one occurrence of w, S_{rij} the set of sentences that contain at least one occurrence of any of the representative words of *(e_j)*, and *N* the number of lines in our corpus. The PMI between (w) and (ej), is exactly the PMI between S_w and S_{rij}, as shown in (1):

$$PMI(S_w, S_{rji}) = \log\left(\frac{card(S_w \cap (\bigcup_{i=1}^{nej} S_{rji})) * N}{card(S_w) * card(\bigcup_{i=1}^{nej} S_{rji})}\right) \quad (1)$$

- In our approach, we used the normalized version of PMI, which gives us the following:

$$NPMI(S_w, S_{rji}) = \frac{PMI(S_w, S_{rji})}{\log(card(S_w \cap (\bigcup_{i=1}^{nej} S_{rji})) / N)} \quad (2)$$

- If a word is part of a negation: it is automatically classified as neutral.
- If we obtain a PMI value equal or lower than 0.25, the corresponding word is classified to "Neutral" category. This threshold is detected through our numerous tests, as the best threshold that allows getting most accurate results.

The algorithm presented is different from the previous version in[4] in many points:

- To calculate the PMI between a word *(w)* and a category of emotion *(e)*, authors in[4] compute first the PMIs between *(w)* and each of the representative word of *(e)*, then the geometric mean of all the PMIs. This method does not consider intersections between the sets S_{rij} of a particular emotion *(e_j)*; such a fact can distort the classification results. To explain this, let us consider the following example: We assume we have two words *w1* and *w2* that we want to classify, and *e* a particular emotion represented by the words r_s. We want to compute the PMIs between *w1*, *w2* respectively and *e* using the old algorithm. Let us consider the case where w1 occurs in few sentences but co-occur with many representative words of *e* in the same sentence. In the other side, the word *w2* occurs in more sentences than *w1*, but it co-occurs with only one representative word of *e* in each sentence. The PMI between *w1* and e could be bigger than *w2* because of co-occurring with many representative words in the same sentence, meaning that w1 is more related to *e* than *w2*. However, the most appropriate is to have *PMI (w2, e)>PMI (w1, e)* since *w2* occurs more in the context of the emotion *e*. The PMI formula we use in our approach is more relevant to computing the relatedness between the two events S_w and S_{rij}. For example, the word "movie" does not express fear, but we could have comments about a scary movie which use a lot of terms that expresses fear in order to describe the movie, which describes the case of *w1= "movie"*. On the other hand, we could have the word "creepy" which occurs in many sentences, but only with one or two representative words of "fear" in each sentence. However, "creepy" is closer than "movie" to the category of emotion "fear".
- The list of representative words in each emotion category is not static like in[4] and can be updated on a regular basis using the former method in order to choose the closest terms to each emotion category. Such detail helps to deal with the evolution of styles of expressions.
- Representative terms of each emotion could be words, but also smiley or regular expressions.
- We use the normalized version of PMI in order to have significant PMI values. The normalized version gives values between -1 and 1. Values that are close to 1 show a strong relatedness. Values close to 0 reflect independence between the compared events. And values close to -1 are given by events that rarely co-occur.

3.2.2. Emotion classification at sentence level

After performing a word level classification of nouns, adjectives, verbs and adverbs that compose the sentence, we have for each word six PMI values corresponding to the target emotions. We compute the averages of PMI values obtained by all the classified words for each emotion category, then, we classify the sentence into the emotion category with the highest average value.

4. Evaluation and results

For this experiment, we have as test set: 138 text entries expressing the six target emotions. They are first stemmed to have the original forms of words, and to make searching possible in our stemmed corpus as detailed above. Our data corpus contains over 200.000 comments retrieved from different YouTube videos. From this data we extract six representative sub-corpuses for the six target emotions. These sub-corpuses are the concrete representation of the union term used in the NPMI formula (1). Each sub-corpus contains only comments using at least one of the representative words of the related emotion. This step helps for a faster computation of all PMIs. We managed to have in our test set, sentences containing affective words like: “Terror held me like a vice-like grip”, and others that do not contain any like: “I was shizzing last night and it burned the bejesus out of me”. As shown in the first table, an entry text is considered as belonging to an emotion class, if the corresponding NPMI is maximal compared to the other emotions.

Table 1. Examples of NPMI values computed by our algorithm.

Joy		Disgust		Anger	
I’m quite excited about my new car!		*Ugh, it's Monday. Barf!*		*I hate it when you're in a crotchety mood*	
disgust: 0.31	**joy : 0.4**	**disgust: 0.34**	joy: 0.0	disgust: 0.29	joy: 0.29
sadness: 0.35	surprise: 0.26	sadness: 0.0	surprise: 0.0	sadness: 0.29	surprise: 0.0
anger: 0.32	fear: 0.26	anger: 0.0	fear: 0.0	**anger: 0.91**	fear: 0.0

In order to evaluate the efficiency of our method, each class of emotion is tested separately. For each emotion we run the algorithm over a sub-test set that contains in its majority sentences that are relevant to the tested emotion, and sentences that are either neutral or expressing other emotions. Testing each emotion separately allows to: first, determine if the method used for building the data corpus, yields a representative dataset for all the target emotions; if we run the algorithm on a representative dataset, we will obtain high precisions for all emotions, because the extracted sub-corpuses will contain enough relevant content for all emotions, which gives us high to chances to find occurrences of any word used in the context of any emotion. Secondly, this allows computing Precision, Recall and Accuracy for each emotion, since these measures focus only on one class. Obtained results are as shown below:

Table 2. Results of emotion classification using our approach.

Emotions	Precision	Recall	Accuracy
Sadness	91.3%	72.41%	67,74%
Joy	91.3%	72.41%	67,74%
Surprise	95.65%	73.33%	70.97%
Anger	95.65%	73.33%	70.97%
Fear	91.3%	72.41%	67,74%
Disgust	91.3%	72.41%	67,74%
Average	**92.75%**	**72.72%**	**68.82%**

We remind that for an emotion *e*, precision, recall and accuracy are as defined below:

$$precision = \frac{tp}{tp + fp} \tag{3}$$

$$recall = \frac{tp}{tp + fn} \tag{4}$$

$$accuracy = \frac{tp + tn}{size\_of\_test\_set} \tag{5}$$

- **Tp:** true positive or number of sentences correctly classified as belonging to e.
- **Tn:** true negative or number of sentences incorrectly classified as belonging to e.
- **Fp:** false positive or number of sentences incorrectly classified as not belonging to e.
- **Fn:** false negative or number of sentences correctly classified as not belonging to e.

As shown in (Table 2), our approach yields an average precision of 92.75%, meaning that this is the percentage of detecting correctly the emotion being expressed in a given text entry. Concerning the accuracy rate, obtained values can be explained by the fact that for each sub-test set, the size of negative sentences is almost the third of the sub-test set size. The second fact behind these results is that we do not have true negative detections, thus the term 'tn' in accuracy measure is equal to zero. Considering these two facts we can say that we have competitive accuracy results.

Previous works such as[12] and[13] obtained, respectively, an average accuracy rate of 71.69 % and 73.89%, while the work of[4], on which we base our approach, yields an average accuracy rate of 57.27%. However, we can't draw a conclusion based on the former measures, for we neither have the same training dataset nor the same test set, as the goal of this approach is to perform emotion classification over text using classical English, as well as, urban expressions and shorthand. Moreover, we do not know precisely the proportions of negative examples in the used test sets. Yet, we can state that our method performs well considering the average rate of precision which outperforms substantially measures given by previous works such as[19] that achieved an average rate of 63.5%.

In summary, our experiments show that YouTube comments provide relevant examples of different emotions expressions, which combined with a statistical classification method, as used in this approach, achieve important accuracy rates. The next step is to extend our range of emotional states, and also elaborate an update system that chooses automatically: the set of representative words for each emotion and the search parameters that achieve best.

5. Conclusion and future work

In this paper, we presented our text-based emotion detection system, based on an unsupervised machine learning algorithm, and using YouTube comments as Data Corpus. The proposed system has two major advantages:

- No Labeling is required in the Data Corpus, which is usually a time consuming task
- The system is flexible enough to allow easy update of the data corpus, and can easily evolve to include new ways of expressions and new concepts.

Our system achieves an average precision of 92.75%, and 68.82% as average accuracy which is close to measures given by previous systems, using SVM as machine learning algorithms.

Future work will focus on extending the range of target emotions, and on considering more linguistic/semantic rules in order to obtain better performance.

References

1. Maglogiannis I, Vouyioukas, Demosthenes Aggelopoulos C. Face detection and recognition of natural human emotion using Markov random fields. *Pers Ubiquitous Comput*. 2009;13:95-101. doi:10.1007/s00779-007-0165-0.
2. Busso C, Metallinou A, S. Narayanan S. ITERATIVE FEATURE NORMALIZATION FOR EMOTIONAL SPEECH DETECTION. In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Prague: IEEE; 2011:5692-5695. doi:10.1109/ICASSP.2011.5947652.
3. Gunes H, Piccardi M. Bi-modal emotion recognition from expressive face and body gestures. *J Netw Comput Appl*. 2007;30(4):1334–1345. doi:10.1016/j.jnca.2006.09.007.
4. Agrawal A, An A. Unsupervised Emotion Detection from Text Using Semantic and Syntactic Relations. In: *2012 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*; 2012:346-353. doi:10.1109/WI-IAT.2012.170.
5. Ekman P. Universals and cultural differences in facial expressions of emotions. In: *Nebraska Symposium on Motivation*.Vol 19; 1972:207–283.
6. Bouma G. Normalized (Pointwise) Mutual Information in Collocation Extraction. *Proc Ger Soc Comput Linguist (GSCL 2009)*. 2009:31-40.
7. Kao EC-C, Liu C-C, Yang T-H, Hsieh C-T, Soo V-W. Towards Text-based Emotion Detection A Survey and Possible Improvements. In: *2009 International Conference on Information Management and Engineering*. Ieee; 2009:70-74. doi:10.1109/ICIME.2009.113.
8. Strapparava C, Valitutti A. WordNet-Affect : an Affective Extension of WordNet. In: *Proceedings of the 4th International Conference on Language Resources and Evaluation*; 2004:1083-1086. doi:10.1.1.122.4281.
9. Stone P, Dunphy D, Smith M, Olgilvie D, et al. The General Inquirer: A Computer Approach to Content Analysis. *J Reg Sci*. 1968;8(1):113–116.
10. Chaumartin F-R. UPAR7: A knowledge-based system for headline sentiment tagging. In: *SemEval 2007 Proceedings of the 4th International Workshop on Semantic Evaluations*; 2007:422-425.
11. Manning C, Surdeanu M, Bauer J, Finkel J, Bethard S, McClosky D. The Stanford CoreNLP Natural Language Processing Toolkit. *Proc 52nd Annu Meet Assoc Comput Linguist Syst Demonstr*. 2014:55-60. Available at: http://www.aclweb.org/anthology/P/P14/P14-5010.
12. Chaffar S, Inkpen D. Using a Heterogeneous Dataset for Emotion Analysis in Text. *ACM Trans Asian Lang Inf Process*. 2006;5(2):165-183. doi:10.1145/1165255.1165259.
13. Aman S, Szpakowicz S. Identifying Expressions of Emotion in Text. In: Matoušek V, Mautner P, eds. *Text, Speech and Dialogue*.Vol 4629; 2007:196-205. doi:10.1007/978-3-540-74628-7_27.
14. Yang C, Lin KH-Y, Chen H-H. Emotion Classification Using Web Blog Corpora. In: *IEEE/WIC/ACM International Conference on Web Intelligence (WI'07)*. Ieee; 2007:275-278. doi:10.1109/WI.2007.51.
15. Lafferty J, McCallum A, Pereira F. "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. 2001.
16. Wu C-H, Chuang Z-J, Lin Y-C. Emotion recognition from text using semantic labels and separable mixture models. *ACM Trans Asian Lang Inf Process*. 2006;5(2):165-183. doi:10.1145/1165255.1165259.
17. DOUIJI Y, Mousannif H. I-CARE: Intelligent Context Aware system for Recognizing Emotions from text. In: *Intelligent Systems: Theories and Applications (SITA)*. Rabat: IEEE; 2015:1 – 5.
18. Mousannif H, Khalil I. *The Human Face of Mobile*. (Linawati, Sudiana Mahendra M, Neuhold EJ, Tjoa AM, You I, eds.). Springer Berlin Heidelberg; 2014.
19. Inkpen D, Keshtkar F, Ghazi D. Analysis and generation of emotion in texts. *Knowl Eng Tech*. 2009:3-14.