Feature Selection for Emotion Classification[*]

Alberto Purpura
purpuraa@dei.unipd.it
University of Padua
Padua, Italy

Chiara Masiero
chiara.masiero@statwolf.com
Statwolf Data Science
Padua, Italy

Gianmaria Silvello
silvello@dei.unipd.it
University of Padua
Padua, Italy

Gian Antonio Susto
sustogia@dei.unipd.it
University of Padua
Padua, Italy

ABSTRACT

In this paper, we describe a novel supervised approach to extract a set of features for document representation in the context of Emotion Classification (EC). Our approach employs the coefficients of a logistic regression model to extract the most discriminative word unigrams and bigrams to perform EC. In particular, we employ this set of features to represent the documents, while we perform the classification using a Support Vector Machine. The proposed method is evaluated on two publicly available and widely-used collections. We also evaluate the robustness of the extracted set of features on different domains, using the first collection to perform feature extraction and the second one to perform EC. We compare the obtained results to similar supervised approaches for document classification (i.e. FastText), EC (i.e. #Emotional Tweets, SNBC and UMM) and to a Word2Vec-based pipeline.

CCS CONCEPTS

• **Information systems → Content analysis and feature selection**; **Sentiment analysis**; • **Computing methodologies → Supervised learning by classification**;

KEYWORDS

Supervised Learning, Feature Selection, Emotion Classification, Document Classification

1 INTRODUCTION

The goal of Emotion classification (EC) is to detect and categorize the emotion(s) expressed by a human. We can find numerous examples in the literature presenting ways to perform EC on different types of data sources such as audio [10] or microblogs [8]. Emotions have a large influence on our decision making. For this reason, being able to understand how to identify them can be useful not only to improve the interaction between humans and machines (i.e. with chatbots, or robots), but also to extract useful insights for marketing goals [7]. Indeed, EC is employed in a wide variety of contexts which include – but are not limited to – social media [8] and online stores – where it is closely related to Sentiment Analysis [9] – with the goal of interpreting emerging trends or to better understand the opinions of customers. In this work, we focus EC approaches which can be applied to textual data. The task is most frequently tackled as a multi-class classification problem. Given

a document d, and a set of candidate emotion labels, the goal is to assign one label to d – sometimes more than one label can be assigned, changing the task to multi-label classification. The most used set of emotions in computer science is the set of the six Ekman emotions [3] (i.e. anger, fear, disgust, joy, sadness, surprise). Traditionally, EC has been performed using dictionary-based approaches, i.e. lists of terms which are known to be related to certain emotions as in ANEW [2]. However, there are two main issues which limit their application on a large scale: (i) they cannot adapt to the context or domain where a word is used (ii) they cannot infer an emotion label for portions of text which do not contain any of the terms available in the dictionary. A possible alternative to dictionary-based approaches are machine learning and deep learning models based on an embedded representation of words, such as Word2Vec [5] or FastText [4]. These approaches however, need lots of data to train an accurate model and they cannot easily adapt to low resource domains. For this reason, we present a novel approach for feature selection and a pipeline for emotion classification which outperform state-of-the-art approaches without requiring large amounts of data. Additionally, we show how the proposed approach generalizes well to different domains. We evaluate our approach on two popular and publicly available data sets – i.e. the Twitter Emotion Corpus (TEC) [6] and SemEval 2007 Affective Text Corpus (1,250 Headlines) [12] – and compare it to state-of-the-art approaches for document representation – such as Word2Vec and FastText – and classification – i.e. #Emotional Tweets [6], SNBC [11] and UMM [1].

2 PROPOSED APPROACH

The proposed approach exploits the coefficients of a multinomial logistic regression model to extract an emotion lexicon from a collection of short textual documents. First, we extract all word unigrams and bigrams in the target collection after performing stopwords removal. [1] Second, we represent the documents using the vector space model (TF-IDF). Then, we train a logistic regressor model with elastic-net regularization to perform EC. This model is characterized by the following loss function:

$$\ell(\{\beta_{0k}, \beta_k\}_1^K) = - \left[\frac{1}{N} \sum_{i=1}^{N} \left(\sum_{k=1}^{K} y_{i\ell}(\beta_{0k} + x_i^T \beta_k) - \log(\sum_{k=1}^{K} e^{\beta_{0k} + x_i^T \beta_k}) \right) \right] \\ + \lambda \left[(1-\alpha)||\beta||_F^2/2 + \alpha \sum_{j=1}^{p} ||\beta||_1 \right],$$

(1)

where β is a $(p+1) \times K$ matrix of coefficients and β_k refers to the k-th column (for outcome category k). For last penalty term $||\beta||_1$, we employ a lasso penalty on its coefficients in order to induce sparse

[1]We employ a list of 170 English terms, see `nltk v.3.2.5` https://www.nltk.org.

Alberto Purpura, Chiara Masiero, Gianmaria Silvello, and Gian Antonio Susto

solution. To solve this optimization problem we use the partial Newton algorithm by making a partial quadratic approximation of the log-likelihood, allowing only (β_{0k}, β_k) to vary for a single class at a time. For each value of λ, we first cycle over all classes indexed by k, computing each time a partial quadratic approximation about the parameters of the current class. [2] Finally, we examine the β-coefficients for each class of the trained model and keep the features (i.e. word unigrams and bigrams) associated to non-zero weights in any of the classes. To evaluate the quality of the extracted features, we perform EC using a Support Vector Machine (SVM). We consider a vector representation of documents based on the set of features extracted as described above, weighting them according to their TF-IDF score.

3 RESULTS

For the evaluation of the proposed approach we consider the TEC and 1,250 Headlines collections. TEC is composed by 21,051 tweets which were labeled automatically – according to the set of six Ekman emotions – using the hashtags they contained and removing them afterwards. We split the collection into a training and a test set of equal size to train the logistic regression model for feature selection. Then, we perform a 5-fold cross validation to train an SVM for EC using the previously extracted features and report in Table 1 the average of the results over all six classes, obtained in the five folds. We also report in Table 1 the performance of FastText – that we computed as in the previous case – and the one of SNBC as described in [11]. From the results in Table 1, we observe that

Method	Mean Precision	Mean Recall	Mean F1 Score
Proposed Approach	**0.509**	0.477	**0.490**
#Emotional Tweets	0.474	0.360	0.406
FastText	0.504	0.453	0.461
SNBC	0.488	**0.499**	0.476

Table 1: Comparison with #Emotional Tweets, FastText and SNBC on the TEC data set.

the proposed classification pipeline outperforms almost all of the selected baselines on the TEC data set. The only exception is SNBC, where we achieve a slighlty lower Recall (-0.022). The 1,250 Headlines data set is a collection of 1,250 newspaper headlines divided in a training (1000 headlines) and a test (250 headlines) set. We employ this data set to evaluate the robustness of the features that we extracted from a randomly sampled subset of tweets equal to 70% of the total size of TEC data set. [3] The results of this experiment are reported in Table 2. We report the performance of (i) a FastText model trained on the training subsed of the data set of 1,000 headlines, (ii) an EC classification pipeline based on Word2Vec and a Gaussian Naive Bayes classifier (GNB) trained on the same training subset of 1,000 headlines of the data set, (iii) #Emotional Tweets, described in [6], and (iv) UMM, reported in [1]. From the results reported in Table 2, we see that our approach outperforms again all the selected baselines in almost all of the evaluations measures. The approach presented in [6] is the only one to have a slightly higher precision than our method (+0.002).

Method	Mean Precision	Mean Recall	Mean F1 Score
Proposed Approach	0.377	**0.790**	**0.479**
FastText	0.442	0.509	0.378
Word2Vec + GNB	0.309	0.423	0.346
#Emotional Tweets	**0.444**	0.353	0.393
UMM (ngrams + POS + CF)	-	-	0.410

Table 2: Comparison with #Emotional Tweets, UMM (best pipeline on the dataset), FastText and Word2Vec+GNB on 250 Headlines data set.

4 DISCUSSION AND FUTURE WORK

We presented and evaluated a supervised approach to perform feature selection for Emotion Classification (EC). Our pipeline relies on a multinomial logistic regression model to perform feature selection, and on a Support Vector Machine (SVM) to perform EC. We evaluated it on two publicly available and widely-used experimental collections, i.e. the Twitter Emotion Corpus (TEC) [6] and SemEval 2007 (1,250 Headlines) [12]. We also compared it to similar techniques such as the one described in #Emotional Tweets [6], FastText [4], SNBC [11], UMM [1] and a Word2Vec-based [5] classification pipeline. We first evaluated our pipeline for EC on documents from the same domain from which the features where extracted (i.e. the TEC data set). Then, we employed it to perform EC on the 1,250 Headlines dataset using the features extracted from TEC. In both experiments, our approach outperformed the selected baselines in almost all the performance measures. More information to reproduce our experiments is provided in [8]. We also make our code publicly available. [4] We highlight that our approach might be applied to other document classification tasks, such as topic labeling or sentiment analysis. Indeed, we are using a general approach adaptable to any task or applicative domain in the document classification field.

REFERENCES

[1] A. Bandhakavi, N. Wiratunga, D. Padmanabhan, and S. Massie. 2017. Lexicon based feature extraction for emotion text classification. *Pattern Recognition Letters* 93 (2017), 133–142.

[2] M. M. Bradley and P. J. Lang. 1999. *Affective norms for English words (ANEW): Instruction manual and affective ratings.* Technical Report. Citeseer.

[3] P. Ekman. 1993. Facial expression and emotion. *American psychologist* 48, 4 (1993), 384.

[4] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov. 2016. Bag of Tricks for Efficient Text Classification. (2016). arXiv:1607.01759 http://arxiv.org/abs/1607.01759

[5] T. Mikolov, I. Sutskever, Chen K., G. S Corrado, and J. Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *NIPS 2013*. 3111–3119.

[6] S. M. Mohammad. 2012. #Emotional tweets. In *Proc. of the First Joint Conference on Lexical and Computational Semantics*. Association for Computational Linguistics, 246–255.

[7] B. Pang and L. Lee. 2008. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval* 2, 1–2 (2008), 1–135.

[8] A. Purpura, C. Masiero, G. Silvello, and G. A. Susto. 2019. Supervised Lexicon Extraction for Emotion Classification. In *Companion Proc. of WWW 2019*. ACM, 1071–1078.

[9] A. Purpura, C. Masiero, and G.A. Susto. 2018. WS4ABSA: An NMF-Based Weakly-Supervised Approach for Aspect-Based Sentiment Analysis with Application to Online Reviews. In *Discovery Science (Lecture Notes in Computer Science)*, Vol. 11198. Springer International Publishing, Cham, 386–401.

[10] F. H. Rachman, R. Sarno, and C. Fatichah. 2018. Music emotion classification based on lyrics-audio using corpus based emotion. *International Journal of Electrical and Computer Engineering* 8, 3 (2018), 1720.

[11] A. G. Shahraki and O. R. Zaiane. 2017. Lexical and learning-based emotion mining from text. In *Proc. of CICLing 2017*.

[12] C. Strapparava and R. Mihalcea. 2007. Semeval-2007 task 14: Affective text. ACL, 70–74.

---

[2]A Python implementation which optimizes the parameters of the model is: https://github.com/bbalasub1/glmnet_python/blob/master/docs/glmnet_vignette.ipynb.
[3]We restricted the training set for the multinomial logistic regressor because of the limitations of the glmnet library we used for its implementation.

[4]https://bitbucket.org/albpurpura/supervisedlexiconextractionforec/src/master/.