Supervised Lexicon Extraction for Emotion Classification

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

Chiara Masiero
chiara.masiero@statwolf.com
Statwolf
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

Emotion Classification (EC) aims at assigning an emotion label to a textual document with two inputs – a set of emotion labels (e.g. anger, joy, sadness) and a document collection. The best performing approaches for EC are dictionary-based and suffer from two main limitations: (i) the out-of-vocabulary (OOV) keywords problem and (ii) they cannot be used across heterogeneous domains. In this work, we propose a way to overcome these limitations with a supervised approach based on TF-IDF indexing and Multinomial Linear Regression with Elastic-Net regularization to extract an emotion lexicon and classify short documents from diversified domains. We compare the proposed approach to state-of-the-art methods for document representation and classification by running an extensive experimental study on two shared and heterogeneous data sets.

CCS CONCEPTS

• Information systems → Content analysis and feature selection; **Sentiment analysis**; **• Computing methodologies → Natural language processing**; **Machine learning**.

ACM Reference Format:
Alberto Purpura, Chiara Masiero, Gianmaria Silvello, and Gian Antonio Susto. 2019. Supervised Lexicon Extraction for Emotion Classification. In *Companion Proceedings of the 2019 World Wide Web Conference (WWW '19 Companion), May 13–17, 2019, San Francisco, CA, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3308560.3316700

1 INTRODUCTION

Emotion Classification (EC) [35] is a fast growing topic in text classification along with Topic Labeling (TL) [34] and Sentiment Analysis (SA) [16]. The research in the field of SA is mainly focused on detecting the subjectivity (objective or subjective) or polarity (positive or negative) of a text rather than specific emotions [20]. EC, on the other hand, is a more fine-grained SA and performs the following task: given a set of emotion labels $\mathcal{E}$ (e.g., anger, joy, sadness, etc.) and a collection of documents $\mathcal{D}$ (e.g., sentences, paragraphs or entire documents), assign to each document a label $e \in \mathcal{E}$. Developing effective techniques to detect emotions in user-generated content can be useful, for instance, to understand the position of a population towards a certain social issue or to assess the success of a marketing campaign [15]. Traditionally, EC has been performed mainly using dictionary-based approaches which employ lists of terms related to specific emotions – e.g. ANEW [3]. Nevertheless, there are two main issues limiting the broad applicability of these approaches: (i) they cannot be employed in domains where a term is used with different emotional connotations than the dictionary denotations; and, (ii) they cannot infer an emotion label for sentences that do not contain any of the known keywords (out-of-vocabulary keyword problem). To overcome these two limitations, we propose a supervised method to extract an emotion lexicon from a given textual corpus to perform EC on other documents from different domains. Our 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 unigrams and bigrams in the chosen collection and consider their TF-IDF weights. Second, we train a logistic regressor to perform EC on the documents of the collection. Third, we create the emotion lexicon by considering all the unigrams and bigrams with non-zero coefficients in the logistic regressor model. We also perform an exhaustive evaluation of the proposed approach. We assess the quality of the selected terms in the lexicon as features for EC by comparing the proposed method to a popular feature selection approach: Principal Component Analysis (PCA). We evaluate the quality of the lexicon for the EC task by employing four supervised classifiers (i.e. K-Nearest Neighbors, Gaussian Naive Bayes, Support Vector Machine, Feed-Forward Neural Network) in order to assess the impact of our lexicon on different classification approaches. We evaluate the generalization power of the lexicon extraction method by employing two heterogeneous public collections (tweets and news headlines). Finally, we compare our lexicon-based document representation approach to Word2Vec [18], a widely-used method to create dense document representations; to FastText [13], a state-of-the-art method for document classification; to a set of Naive Bayes classifiers (SNBC), each trained to recognize one emotion, as done in [29]; and to the approach presented in [1], which employs a generative unigram mixture model (UMM) to model emotionality and neutrality of words from labeled documents. The main contributions of this work are:

WWW '19 Companion, May 13–17, 2019, San Francisco, CA, USA

ACM ISBN 978-1-4503-6675-5/19/05.
https://doi.org/10.1145/3308560.3316700

- the application of a method based on regularized multinomial logistic regression to build an emotion lexicon;
- the feature selection process of the lexicon extraction method which employs: (i) unsupervised methods for document representation (TF-IDF and Word2Vec); and, (ii) supervised and unsupervised methods for dimensionality reduction (logistic regression and PCA);
- the evaluation of the discriminative power of the selected features by means of extensive classification experiments with four different classifiers of increasing complexity;
- the comparison of our best performing classification pipeline to #Emotional Tweets [19] – i.e., the best-known baseline on the Twitter Emotion Corpus (TEC) data set;
- the comparison of our best performing classification pipeline to SNBC [29] and UMM [1], two other recent approaches for emotion classification;
- a "transfer learning" experiment to assess the consistency of the proposed approach across different domains. We extract the lexicon from the TEC data set and we test it on the TEC itself and on the SemEval 2007 Affective Text Corpus [31] (1000 and 250 News Headlines) for the EC task;
- a shared public repository containing the open source code and references to the data to reproduce our experiments.

We show that the proposed approach performs better overall than the chosen baselines (i.e., #Emotional Tweets, UMM, SNBC) both on the homogeneous scenario (learn and test on tweets) and on the heterogeneous scenario (learn on tweets and test of news headlines). Moreover, we show that our approach for short documents representation is highly competitive with two cornerstone methods like Word2Vec and FastText.

The rest of the paper is organized as follows: in Section 2 we give an overview of the most popular strategies for EC; in Section 3 we present our approach for EC and its main components; in Section 4 we describe the shared collections used in our experiments and our experimental setup; in Section 5, we present the results of different classification systems. Finally, in Section 6 we draw some conclusions and outline future work.

2 RELATED WORK

In the last years, SA and Opinion Mining research increasingly focused on the classification of user-created contents like tweets [6, 21], news [14] or customer reviews [25] available on the Web. Despite this, the importance of a more fine-grained emotion classification (EC) on this type of documents has emerged only recently [12].

There are two main approaches to EC: the categorical one [5], which consists of assigning a label to each element to be classified; and the dimensional one [27], which attempts to represent detected emotions in a space, for example of two dimensions, such as valence (i.e. pleasure/displeasure) and arousal (i.e. activation/deactivation). In the literature, EC has been applied to different domains and textual data of different lengths, from posts on social media [35] to fairy tales [10]. Each application domain has its own peculiarities. Specifically, classification for shorter texts is usually more challenging and less effective. In this work, we focus on the analysis of short texts and on the problem of how to extract meaningful features for the classification of emotions.

Some approaches tackle emotion classification as a multi-label classification problem. In [4] for example, the authors assign zero or more emotion labels to each sentence in a movie review and propose a method to learn dependencies between labels and exploit this information during the classification. In this work, we consider the problem as a multi-class classification problem since this is the most widespread formulation, and it allows us to evaluate more easily our method of emotion lexicon extraction. This is a reasonable assumption since in our case we are dealing with short texts which usually express a single emotion.

Different sets of 6, 8 or 20 emotions [7, 11, 23, 24] have been considered in the literature; however, the set of six Ekman emotions [7] (anger, disgust, fear, joy, sadness, and surprise) has become the most popular choice in many studies such as in [32] and in SemEval 2007 "Affective Text" task [31]. For this reason, we employ this set of six emotions to evaluate our approach and compare it to #Emotional Tweets [19] where the author propose another supervised method to extract features for EC in tweets based on Pointwise Mutual Information (PMI). In [19], the author employs PMI to select word n-grams according to their correlation with emotion labels in a data set, then uses the information on the presence or absence of these features in a document to perform its classification using a Support Vector Machine (SVM) classifier.

The original approach presented in this work for document representation, dimensionality reduction and emotion lexicon extraction is based on a multinomial regression model which is employed to select the most relevant features for the classification task. To the best of our knowledge, we are the first to apply, in the EC domain, a multinomial regression model for lexicon extraction.

A related approach to ours is FastText [13], a state-of-the-art method for document classification. FastText first learns a word embedding, then averages word representations into a text representation, and finally uses softmax to compute the probability distribution on the predefined classes. This architecture is inspired to the CBOW Word2Vec [18] model. In particular, in FastText the middle word used in CBOW is replaced by a class label. This approach, similarly to what we propose in this work, uses a labeled data set to create document embeddings with discriminant features and has been successfully used for SA before [22]. For this reason, we compare our approach for EC also to FastText.

We also compare our approach to two other recent approaches for emotion classification. SNBC [29], which employs a set of Naive Bayes classifiers (SNBC) – each trained to recognize one emotion – for emotion classification; and to the approach presented in [1], which employs a generative unigram mixture model (UMM) to model emotionality and neutrality of words from labeled documents and then classify them.

3 PROPOSED APPROACH

We propose to use a labeled collection of documents where each item is associated with an emotion label and begin by indexing it with TF-IDF, extracting all the unigrams and bigrams which appear in the collection more than 5 times. [1] Then, we train a multinomial logistic regression model with elastic-net regularization and consider its sparse coefficients matrix β (see Equation 4). For each class, we keep the features extracted with TF-IDF, which have non-zero coefficients in the respective column of β. This set of unigrams and bigrams constitutes the emotion lexicon we extract for each emotion.

The components of our lexicon extraction and evaluation pipeline are detailed in the rest of this section.

3.1 Emotion Lexicon Extraction

Document Representation. We first perform stopwords removal; [2] secondly, we create document embeddings using TF-IDF indexing [17]. TF-IDF is a well-established technique and at this step in our pipeline we index the documents considering all unigrams and bigrams in the collection.

Lexicon Extraction. The proposed approach for lexicon extraction is based on *multinomial logistic regression* (MLR) with *elastic net regularization* [2]. Logistic regression is a non-linear model for classification, also known as *logit regression, maximum-entropy (MaxEnt) or log-linear classification.* This model estimates the probabilities describing the possible outcomes of the classifier using a logistic function. For binary classification problems, suppose we have a response variable that takes values in $G = \{0, 1\}$; we denote with $y_i = I(g_i = 1)$ the indicator response variable. We model the logistic function as

$$\Pr(G = 1|X = x) = \frac{e^{\beta_0 + \beta^T x}}{1 + e^{\beta_0 + \beta^T x}}, \tag{1}$$

where β is a p-dimensional array and p is the size of an input sample, β_0 is a scalar and represents the intercept of the model. The objective function for the elastic net penalized logistic regression uses the negative binomial log-likelihood (where $\alpha \in [0, 1]$ is the elastic-net regularization coefficient):

$$\begin{aligned} \min_{(\beta_0,\beta)\in\mathbb{R}^{p+1}} & \lambda[(1-\alpha)||\beta||_2^2/2 + \alpha||\beta||_1] - \\ & \left[\frac{1}{N}\sum_{i=1}^{N} y_i(\beta_0 + x_i^T\beta) - \log(1 + e^{\beta_0 + x_i^T\beta})\right]. \end{aligned} \tag{2}$$

For multi-class classification problems with K classes [30], the logistic function is

$$\Pr(G = k|X = x) = \frac{e^{\beta_{0k} + \beta_K^T x}}{\sum_{\ell=1}^{K} e^{\beta_{0\ell} + \beta_\ell^T x}}, \tag{3}$$

where the response variable has K levels $\mathcal{G} = \{1, 2, ..., K\}$. Let Y be the $N \times K$ indicator response matrix, with elements $y_{i\ell} = I(g_i = \ell)$. Then, the elastic net penalized negative log-likelihood function [33] is

$$\begin{aligned} & \ell(\{\beta_{0k}, \beta_k\}_1^K) = \\ & -\left[\frac{1}{N}\sum_{i=1}^{N}\left(\sum_{k=1}^{K} y_{i\ell}(\beta_{0k} + x_i^T\beta_k) - \log(\sum_{k=1}^{K} e^{\beta_{0k} + x_i^T\beta_k})\right)\right] \\ & + \lambda\left[(1-\alpha)||\beta||_F^2/2 + \alpha\sum_{j=1}^{p}||\beta||_1\right], \end{aligned} \tag{4}$$

where β is a $(p + 1) \times K$ matrix of coefficients, β_k refers to the k-th column (for outcome category k) and β_j the j-th row (vector of K coefficients for variable j). The last penalty term is $||\beta||_1$ where we employed a lasso penalty on its coefficients in order to induce sparse 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. [3] After obtaining a model, we examine the β-coefficients for each class and keep the non-zero features extracted with TF-IDF. Finally, for EC, we consider the terms associated to non-zero coefficients and we index the documents in the experimental collections considering only these terms with their respective TF-IDF weight.

3.2 Emotion Lexicon Evaluation

In order to evaluate the quality of the features selected with our approach for lexicon extraction, we compare it to another method for feature selection and dimensionality reduction: *Principal Component Analysis* (PCA) [36]. We use PCA to decompose a multivariate data set in a set of successive orthogonal components that explain a maximum amount of the variance. In this case, our goal is to create a new lower-dimensional set of features to represent a textual document in order to later detect the emotions expressed within the document itself. We employed the randomized truncated Singular Value Decomposition (SVD) [4] to create new document embeddings of size 50 (this size leads to the best performances on both the test data sets as we have assessed empirically).

To check if the proposed approach for feature selection leads to a consistent performance improvement because of the discriminant power of the extracted lexicon, we perform EC using different classifiers of increasing complexity.

We classify the documents according to the emotions they express in a multi-class fashion. For data sets which associate more than one label to each document, we train a classifier for each class (i.e. to predict the labels "Joy" and "Not-Joy"), repeat the process for each emotion, and solve the multi-label classification problem as a set of binary classification problems. The classifiers we employ are the following:

- *K-Nearest Neighbors* (K-NN): a simple widely-known supervised classifier based on the distance between elements in

[1] The choice of this frequency threshold is based on the results of the experiments on the selected collections.

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

[3] A Python implementation which optimizes the parameters λ and α of the model is: https://github.com/bbalasub1/glmnet_python/blob/master/docs/glmnet_vignette.ipynb.

[4] We used the implementation available in scikit-learn library v.0.19.1 for Python http://scikit-learn.org/stable/.

the input space. Different distance metrics can be used, we considered the most widely-used Euclidean distance;

- *Gaussian Naive Bayes* (GNB): a supervised probabilistic classifier based on the Bayesian model [28], which assumes independence between the input features and does not require any hyper-parameter optimization;
- *Support Vector Machine* (SVM): one of the most popular and effective supervised classifiers [28]. This is also the classifier employed in #Emotional Tweets, the baseline to which we compare the proposed approach. In our experiments, we decided to employ an SVM with a linear kernel in order to keep the model as simple as possible and to speed up the optimization process of its hyper-parameters;
- *Feed-Forward Neural Network* (FFNN): a supervised classifier which can be of arbitrary complexity growing with the number of layers and hidden units of the network. We included this classifier in our pipeline since Neural Network and Deep Learning-based approaches are attracting great interest in the NLP and, more in general, in the Machine Learning community. The hyper-parameters that we considered for the optimization of this classifiers are the number of layers and their size. We considered different combinations of one or two layers of sizes from 5 to 5000.

In order to evaluate the generalization power of the proposed approach for lexicon extraction, we extract an emotion lexicon from the TEC data set and then perform EC on the SemEval 2007 Affective Text Corpus. The results of our evaluation are reported in Section 5.

Finally, we compare the results of different classification pipelines to #Emotional Tweets, SNBC, UMM, Word2Vec and FastText as alternative and effective methods for dense document representation.

4 EXPERIMENTAL SETUP AND REPRODUCIBILITY

The implementation of the proposed methods and the code for their evaluation is available on our public repository. [5] For the evaluation of our pipelines we employ two publicly available data sets:

- 1000 and 250 Headlines [6]: the SemEval 2007 Affective Text corpus [31] contains 1250 newspaper headlines (1000 for training and 250 for testing) labeled with the six Ekman emotions by six annotators. For each headline-emotion pair, the annotators assigned scores from 0 to 100 indicating how strongly the emotion was expressed in the headline. For our experiments, like in [19], we considered scores greater or equal to 25 to indicate that the headline expresses the corresponding emotion;
- TEC [7]: the Twitter Emotion Corpus (TEC) data set [19] contains a set of 21,051 tweets labeled with the six Ekman emotions downloaded with the Twitter API [8] and labeled according to their hashtags (i.e., #anger, #disgust, #fear, #joy, #sadness, #surprise).

We begin by training a Word2Vec model [9] to obtain vectors with 100 features by employing the skip-gram algorithm with negative sampling (5 negative samples), a window of 5 terms, and we filter out words that appear in the collection less than 10 times. We iterate the training for 5 epochs on each of the evaluation data sets. Document embeddings are obtained by averaging their word embeddings. Then, for our experiments, we first extract (for each data set) the features in an unsupervised way with TF-IDF or Word2Vec (or in a supervised way with FastText [10]); after this, we reduce the size of the TF-IDF embeddings by training the multinomial logistic regression model on the data set split into a training and a test set of equal size; finally, we apply the classifiers we described in Section 3.2. In order to statistically validate our results, we employ a 5 folds cross-validation procedure for each combination of the our proposed classification pipelines. Since the scores relative to each fold of the baseline method [19] are not available, we are not able to compute any statistical test to check if our results are statistically different from the chosen baseline; nevertheless, we report a comparison of the metrics obtained by summing and then averaging the results from each run.

The hyper-parameters of the considered classifiers have been obtained by optimizing them using the scikit-learn RandomizedSearchCV class. [11] For all our experiments we decided to keep the optimized configuration of the classifiers hyper-parameters associated to the documents embeddings created with TF-IDF on TEC data set. We also evaluated the impact of the optimization of the classifiers on the 1000 Headlines data set and noticed no relevant performance improvement with a different configuration of the hyper-parameters.

The hyper-parameters employed for each document classifier [12] are the following:

- K-NN: 2 neighbors;
- GNB: no hyper-parameters required;
- SVM: linear kernel with error penalty $C = 12.5$ on the default l_2 norm;
- FFNN: a single-layer Feed-Forward Neural Network of size 2000.

The emotion lexicon we employ has been extracted from a subset of the TEC data set (70%) randomly sampled without replacement. [13] The performance measures we employed for the evaluation of our approach on the Headlines data set displayed in Table 2 are mean precision, mean recall and mean f1 score calculated as the average of single class precision (recall, f1 score) over all considered classes. We obtained the measures relative to FastText in Table 2 converting the class probabilities returned by the algorithm to binary values according to a threshold value. We decided to evaluate FastText on the best-case scenario, selecting the threshold which led to the best results on the test data set.

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

[6] http://web.eecs.umich.edu/~mihalcea/affectivetext/.

[7] https://bit.ly/2M0KCku.

[8] https://developer.twitter.com/en/docs/tweets/search/overview.

[9] Gensim library for Python [26].

[10] https://fasttext.cc.

[11] http://scikit-learn.org/stable/modules/generated/sklearn.model_selection.RandomizedSearchCV.html.

[12] The implementation of each classifier can be found in the scikit-learn library v.0.19.1 http://scikit-learn.org/stable/.

[13] This is due to some limitations when dealing with large quantities of data of the `glmnet` library we use to train the regression model.

5 EVALUATION

5.1 Features Quality

In Figure 1 and Table 1, we report the results obtained on TEC data set. In this case, the problem to solve is single label multi-class classification. We evaluated four different classification pipelines and report the F1 score for each pipeline and emotion:

- TF-IDF: we employed TF-IDF indexing to compute the document embeddings and we classified them using four different classifiers;
- TF-IDF w/MLR: we employed a Multinomial Logistic Regressor (MLR) to extract an emotion lexicon for each class and then performed document classification using the TF-IDF embeddings relative to that lexicon;
- TF-IDF w/PCA: we employed TF-IDF indexing to compute the document embeddings and then reduced their size using PCA. We report the classification performance using four different classifiers;
- W2V: we computed document embeddings by averaging the term embeddings of each document and classified them using four different classifiers;
- ET: the #Emotional Tweets baseline on the TEC data set performance (obtained with an SVM classifier);
- MLR: we employed TF-IDF indexing to compute the document embeddings, then employed a MLR for the classification of the documents, without any feature selection step;
- SNBC: we report the results obtained on the TEC dataset by [29] with a set of Naive Bayes classifiers, each trained to predict one emotion, using unigrams and bigrams as lexical features.

Based on the charts in Figure 1 (and in Table 1 for a more accurate analysis), we can make five conclusions. First, dimensionality reduction (either with PCA or with the selected lexicon) leads generally to a performance improvement in the EC task. Second, the proposed approach for feature selection leads to a stable performance increase with almost all of the classifiers (the GNB classifier is the only exception on the TEC dataset). Third, if we compare the proposed approach for document representation to the pipeline using Word2Vec, we notice a better performance of our approach with all of the classifiers except for the GNB. Fourth, we notice that when we combine our method for lexicon extraction with the SVM classifier (or the FFNN), as it is done in #Emotional Tweets, we always obtain a better performance than the current baseline on the TEC data set. Finally, we observe that the F1 scores of the proposed approach for feature selection, when used in the same pipeline of the FFNN or SVM classifiers, are always higher or equal to the cases where only a MLR was employed for the classification, without feature selection.

We compare our approach to FastText – considered here as another state-of-the-art approach for document classification – and our method outperforms it in several cases; FastText performs better than our approach only on precision for anger and disgust and on F1 score for sadness and surprise. It is worth noticing that an advantage of our approach with respect to FastText is the *interpretability* of predictions. In fact, FastText is based on an artificial neural network and it is difficult to figure out which features play a major role in classification. On the contrary, in our pipeline we have access to the weights relative to each unigram or bigram selected by the MLR model and the SVM classifier. Hence, we can estimate the influence of each feature on the classification of each document. Moreover, our model does not require to learn an embedding for unigrams and bigrams, differently from FastText. Thus, it is more suitable for classification over small collections.

We also compare our approach to SNBC [29], and we outperform it in most of the cases with just one exception in the classification of tweets expressing anger.

Therefore, the proposed approach for lexicon extraction is a trustworthy and well-performing new method to select terms in a corpus which are discriminative for the classification and it outperforms the current baselines.

5.2 Generalization Power and Overall Performance

In Table 2 we compare our approach for emotion lexicon extraction with the method proposed in #Emotional Tweets, a Word2Vec-based document representation, FastText, and the UMM approach presented in [1]. We extract an emotion lexicon (unigrams and bigrams) from TEC data set then, we evaluate the classification accuracy with the lexicon extracted from the training data set on the 250 Headlines collection. [14] We employed this experimental setup to be able to compare our approach to the chosen baseline system [19] under the same conditions. Contrarily to [19] – for the hardware limitations of our setup – we employ only 70% of TEC data set for lexicon extraction instead of the whole collection, then perform the evaluation on the 250 Headlines collection. For the experiments using Word2Vec and FastText, we trained, respectively, the GNB classifier and the FastText model, on the 1000 Headlines collection.

As we can see in Table 2, our method for the extraction of an emotion lexicon outperforms the other approaches and the baseline in terms of recall and F1 score.

Moreover, we observe that the number of features in the emotion lexicon extracted with our approach is roughly half of those of the baseline; this indicates that we are selecting a more restricted subset of elements, without compromising the overall classification quality.

In the comparison with UMM, we considered its best performing case, where additional information was used and training and testing were performed on data from the same domain. The authors employed the following additional features: Part-Of-Speech (POS) tags and Contextual Features (CF) which include punctuation marks, emoticons, capitalized words, elongated words, negations and sentiment features. [15] Even though in [1] the authors indicate only the average F1 score of their evaluation results, we see that our approach – even in a cross-domain classification scenario – outperforms UMM.

[14] We used the 1000 Headlines collection as a training set for the classifier as done in the baseline system.

[15] For a complete description of these features, please refer to [1]

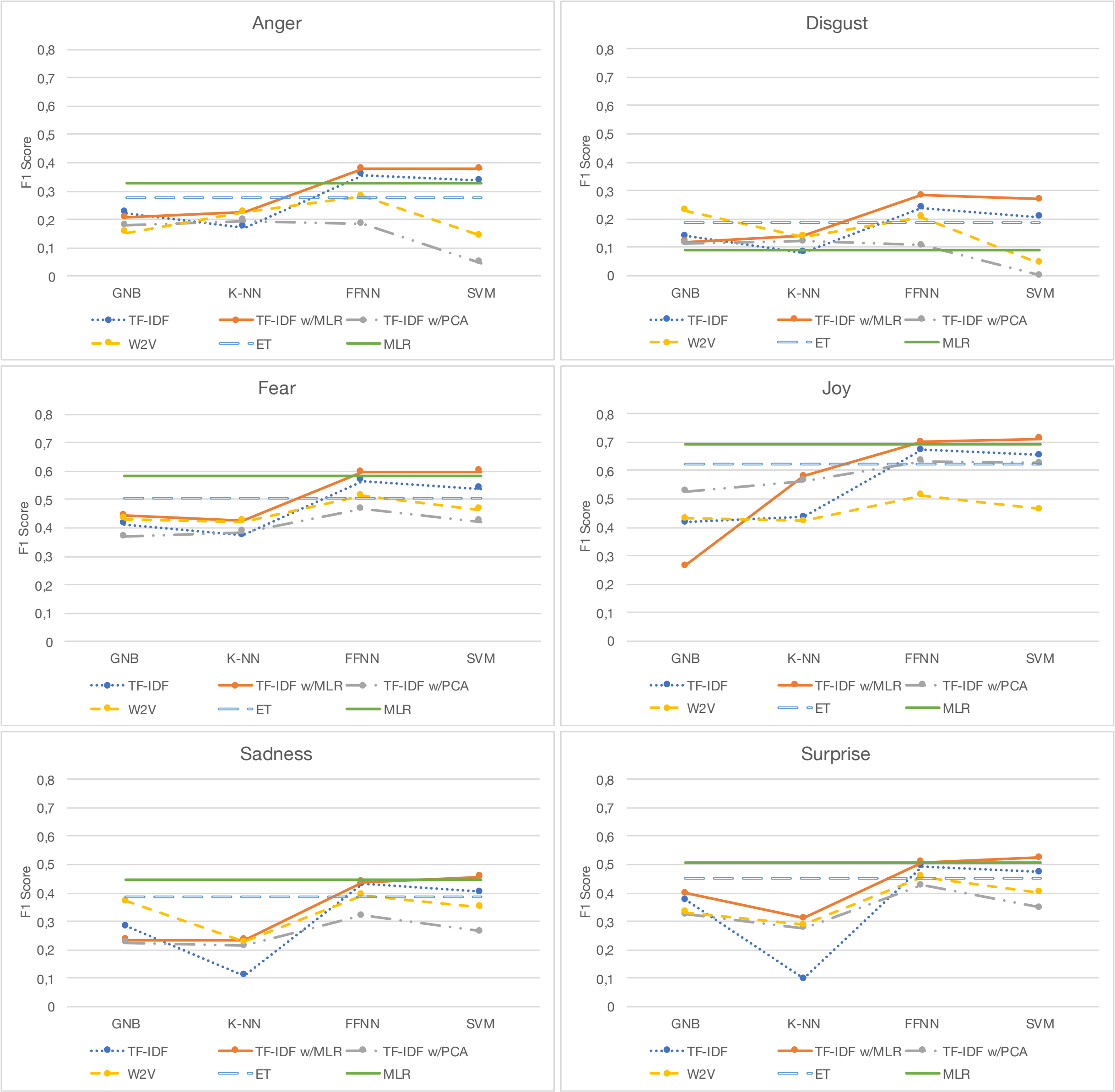

Figure 1: Classification results on TEC. The F1 scores were computed considering the sum of the total number of documents belonging to the class, the total number of correct predictions and the total number of performed predictions for each class.

6 CONCLUSIONS AND FUTURE WORK

In this work, we proposed and evaluated a novel approach for emotion lexicon extraction. The process employs a multinomial logistic regressor to extract an emotion lexicon from a labeled collection of documents.

To evaluate the quality of the extracted lexicon, we performed two different tests: (i) we considered lexicon extraction as a feature selection/dimensionality reduction problem and compared it to PCA; (ii) we compared the classification performance of different classifiers with the proposed lexicon-based document representation in order to assess the discriminant power of the lexicon terms.

The results of the above experiments showed a consistent performance improvement in the EC task when employing the proposed approach. We also compared our approach for document representation to Word2Vec, considered as the most widespread alternative to obtain dense document representations, to FastText, a state-of-the-art method for document classification, and to SNBC [29]. With respect to Word2Vec and SNBC, we obtained better performances for almost all emotions (with the exception of the emotion anger, where SNBC is the best option), and we are competitive also with FastText, which outperforms our approach only in a handful of cases. Furthermore, the proposed approach for EC has the advantage of being more interpretable than embedding-based ones since the features we used to perform the classification are easily accessible and represented by word unigrams and bigrams. Finally, we evaluated the generalization power of the lexicon extraction process by generating a lexicon from TEC data set, and using that for the classification of the documents in the SemEval 2007 Affective Text corpus. Also in this case, our approach performed better overall than #Emotional Tweets [19] and UMM [1] the baseline systems chosen as reference.

All the information necessary to reproduce our experiments, including the details about the training of the classifiers, is provided. We also make our code publicly available.

We highlight that our approach might be applied to document classification also for other 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. Another element of investigation will be the analysis of the impact of the pre-processing step for document representation. Finally, we plan to conduct a thorough statistical analysis by means of general linear mixed models in order to determine the contribution of each

Table 1: Classification results on TEC data set (full results in Figure 1). We highlight in bold the top performance value of each measure for each emotion.

Emotion	Doc. Repr.	Dim. Red.	Classifier	Prec.	Rec.	F1 Score	**Emotion**	Doc. Repr.	Dim. Red.	Classifier	Prec.	Rec.	F1 Score
anger	TF-IDF	MLR	GNB	0.137	0.428	0.208	**joy**	TF-IDF	MLR	GNB	**0.734**	0.161	0.264
			K-NN	0.179	0.309	0.226				K-NN	0.532	0.637	0.580
			FFNN	0.395	0.365	**0.379**				FFNN	0.655	0.751	0.700
			SVM	0.418	0.344	0.377				SVM	0.657	**0.777**	**0.712**
		none	GNB	0.162	0.356	0.222			none	GNB	0.618	0.316	0.418
			K-NN	0.148	0.209	0.173				K-NN	0.526	0.374	0.437
			FFNN	0.369	0.348	0.358				FFNN	0.659	0.687	0.672
			SVM	0.339	0.333	0.336				SVM	0.642	0.666	0.654
		PCA	GNB	0.113	0.440	0.180			PCA	GNB	0.591	0.474	0.526
			K-NN	0.147	0.291	0.195				K-NN	0.542	0.586	0.563
			FFNN	0.247	0.148	0.185				FFNN	0.556	0.735	0.633
			SVM	0.328	0.025	0.047				SVM	0.479	0.899	0.625
	W2V	none	FFNN	0.312	0.254	0.280		W2V	none	FFNN	0.636	0.681	0.658
	ET	none	SVM	0.373	0.223	0.279		ET	none	SVM	0.645	0.604	0.624
	FastText	none	none	**0.422**	0.233	0.300		FastText	none	none	0.654	0.765	0.705
	SNBC	none	none	0.304	**0.452**	0.363		SNBC	none	none	0.72	0.691	0.705
disgust	TF-IDF	MLR	GNB	0.067	0.522	0.118	**sadness**	TF-IDF	MLR	GNB	0.332	0.181	0.235
			K-NN	0.131	0.152	0.141				K-NN	0.328	0.183	0.235
			FFNN	0.327	0.251	**0.284**				FFNN	0.457	0.422	0.439
			SVM	0.300	0.244	0.270				SVM	**0.484**	0.435	0.458
		none	GNB	0.097	0.259	0.141			none	GNB	0.277	0.290	0.283
			K-NN	0.045	**0.530**	0.082				K-NN	0.388	0.063	0.109
			FFNN	0.274	0.213	0.239				FFNN	0.437	0.425	0.431
			SVM	0.217	0.200	0.208				SVM	0.412	0.395	0.403
		PCA	GNB	0.076	0.230	0.114			PCA	GNB	0.332	0.170	0.225
			K-NN	0.092	0.179	0.121				K-NN	0.296	0.169	0.215
			FFNN	0.170	0.079	0.108				FFNN	0.356	0.292	0.321
			SVM	0.000	0.000	0.000				SVM	0.378	0.205	0.266
	W2V	none	FFNN	0.250	0.176	0.207		W2V	none	FFNN	0.402	0.385	0.393
	ET	none	SVM	0.307	0.134	0.187		ET	none	SVM	0.419	0.360	0.387
	FastText	none	none	**0.350**	0.200	0.255		FastText	none	none	0.467	0.510	0.488
	SNBC	none	none	0.171	0.427	0.243		SNBC	none	none	0.470	**0.517**	**0.492**
fear	TF-IDF	MLR	GNB	0.391	0.508	0.442	**surprise**	TF-IDF	MLR	GNB	0.485	0.339	0.399
			K-NN	0.350	0.537	0.424				K-NN	0.641	0.205	0.311
			FFNN	0.619	**0.573**	0.595				FFNN	0.538	0.480	0.508
			SVM	**0.642**	0.562	**0.599**				SVM	0.551	0.500	0.524
		none	GNB	0.349	0.509	0.414			none	GNB	0.411	0.348	0.377
			K-NN	0.358	0.389	0.373				K-NN	**0.706**	0.053	0.099
			FFNN	0.569	0.562	0.565				FFNN	0.491	0.497	0.494
			SVM	0.543	0.536	0.539				SVM	0.475	0.473	0.474
		PCA	GNB	0.489	0.298	0.370			PCA	GNB	0.423	0.265	0.326
			K-NN	0.323	0.476	0.385				K-NN	0.568	0.184	0.278
			FFNN	0.511	0.430	0.467				FFNN	0.457	0.400	0.426
			SVM	**0.642**	0.314	0.422				SVM	0.511	0.265	0.349
	W2V	none	FFNN	0.536	0.490	0.512		W2V	none	FFNN	0.439	0.478	0.458
	ET	none	SVM	0.596	0.439	0.506		ET	none	SVM	0.506	0.405	0.450
	FastText	none	none	0.569	0.510	0.488		FastText	none	none	0.561	**0.502**	**0.530**
	SNBC	none	none	0.634	0.503	0.561		SNBC	none	none	0.626	0.403	0.489

Table 2: Comparison with #Emotional Tweets and UMM [1] (best pipeline on the dataset) on the task of lexicon extraction from the TEC data set, evaluated on 250 Headlines data set, considering the best performing pipelines.

Method	# of Features extracted from TEC lexicon	Mean Precision	Mean Recall	Mean F1 Score
#Emotional Tweets	11,418	**0.444**	0.353	0.393
TF-IDF+MLR+GNB	6,383	0.377	**0.790**	**0.479**
Word2Vec + GNB	100 (document embeddings size)	0.309	0.423	0.346
FastText	100 (document embeddings size)	0.442	0.509	0.378
UMM (ngrams + POS + CF)	-	-	-	0.410

component to the overall pipeline similarly to what has been done for off-the-shelf information retrieval systems in [8, 9].

REFERENCES

[1] Anil Bandhakavi, Nirmalie Wiratunga, Deepak Padmanabhan, and Stewart Massie. 2017. Lexicon based feature extraction for emotion text classification. *Pattern recognition letters* 93 (2017), 133–142.

[2] Christopher Bishop. 2016. *Pattern recognition and machine learning.* Springer-Verlag, New York.

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

[4] Lars Buitinck, Jesse Van Amerongen, Ed Tan, and Maarten de Rijke. 2015. Multi-emotion detection in user-generated reviews. In *Proc. of the 37th European Conference on IR Research, ECIR 2015 (LNCS)*, Vol. 9022. Springer, 43–48.

[5] Rafael A Calvo and Sunghwan Mac Kim. 2013. Emotions in text: dimensional and categorical models. *Computational Intelligence* 29, 3 (2013), 527–543.

[6] Keith Cortis, André Freitas, Tobias Daudert, Manuela Huerlimann, Manel Zarrouk, Siegfried Handschuh, and Brian Davis. 2017. Semeval-2017 task 5: Fine-grained sentiment analysis on financial microblogs and news. In *Proc. of the 11th International Workshop on Semantic Evaluation (SemEval-2017).* 519–535.

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

[8] Nicola Ferro and Gianmaria Silvello. 2016. A General Linear Mixed Models Approach to Study System Component Effects. In *Proc. 39th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2016).* ACM Press, New York, USA.

[9] Nicola Ferro and Gianmaria Silvello. 2017. Towards an Anatomy of IR System Component Performances. *Journal of the American Society for Information Science and Technology (JASIST)* 69, 2 (2017), 187–200.

[10] Virginia Francisco and Pablo Gervás. 2006. Automated mark up of affective information in english texts. In *Proc. of the 9th International Conference on Text, Speech and Dialogue, TSD 2006 (LNCS)*, Vol. 4188. Springer, 375–382.

[11] Nico H Frijda. 2017. *The laws of emotion.* Psychology Press.

[12] Mohammed Jabreel and Antonio Moreno. 2018. EiTAKA at SemEval-2018 Task 1: An Ensemble of N-Channels ConvNet and XGboost Regressors for Emotion Analysis of Tweets. *arXiv preprint arXiv:1802.09233* (2018).

[13] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of Tricks for Efficient Text Classification. (2016). arXiv:1607.01759 http://arxiv.org/abs/1607.01759

[14] Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Recurrent Convolutional Neural Networks for Text Classification.. In *Proc. of the 29th AAAI Conference on Artificial Intelligence*, Vol. 333. AAAI Press, 2267–2273.

[15] Bing Liu. 2012. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies* 5, 1 (2012), 1–167.

[16] Bin Lu, Myle Ott, Claire Cardie, and Benjamin K Tsou. 2011. Multi-aspect sentiment analysis with topic models. In *IEEE 11th International Conference on Data Mining Workshops (ICDMW 2011).* IEEE Computer Society, 81–88.

[17] James Mayfield and Paul McNamee. 1998. Indexing using both n-grams and words. In *Proc. of the 7th Text REtrieval Conference, TREC 1998*, Vol. Special Publication 500-242. National Institute of Standards and Technology (NIST), 361–365.

[18] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *Proc. of the 27th Annual Conference on Neural Information Processing Systems (NIPS 2013).* 3111–3119.

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

[20] Myriam D Munezero, Calkin Suero Montero, Erkki Sutinen, and John Pajunen. 2014. Are they different? Affect, feeling, emotion, sentiment, and opinion detection in text. *IEEE transactions on affective computing* 5, 2 (2014), 101–111.

[21] Preslav Nakov, Alan Ritter, Sara Rosenthal, Fabrizio Sebastiani, and Veselin Stoyanov. 2016. SemEval-2016 task 4: Sentiment analysis in Twitter. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016).* 1–18.

[22] Leonard Papenmeier and Christoph Friedrich. 2017. Fasttext and Gradient Boosted Trees at GermEval-2017 on Relevance Classification and Document-level Polarity. In *Proceedings of the GSCL GermEval Shared Task on Aspect-based Sentiment in Social Media Customer Feedback.* 30–35.

[23] W Gerrod Parrott. 2001. *Emotions in social psychology: Essential readings.* Psychology Press.

[24] Robert Plutchik and Henry Kellerman (Eds.). 1980. *Theories of Emotion (Volume 1): Theory, Research, and Experience.* New York: Academic Press.

[25] Alberto Purpura, Chiara Masiero, and Gian Antonio Susto. 2018. WS4ABSA: an NMF-based Weakly-Supervised Approach for Aspect-Based Sentiment Analysis with Application to Online Reviews. In *Proc. of the 24th International Syposium on Methodologies for Intelligent Systems (ISMIS 2018).* Springer.

[26] Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proc. of the LREC 2010 Workshop on New Challenges for NLP Frameworks.* ELRA, 45–50.

[27] James A Russell. 2003. Core affect and the psychological construction of emotion. *Psychological review* 110, 1 (2003), 145.

[28] Stuart J Russell and Peter Norvig. 2016. *Artificial Intelligence: A Modern Approach.* Pearson Education Limited.

[29] Ameneh Gholipour Shahraki and Osmar R Zaiane. 2017. Lexical and learning-based emotion mining from text. In *Proceedings of the International Conference on Computational Linguistics and Intelligent Text Processing.*

[30] Noah Simon, Jerome Friedman, and Trevor Hastie. 2013. A blockwise descent algorithm for group-penalized multiresponse and multinomial regression. *arXiv preprint arXiv:1311.6529* (2013).

[31] Carlo Strapparava and Rada Mihalcea. 2007. Semeval-2007 task 14: Affective text. Association for Computational Linguistics, 70–74.

[32] Carlo Strapparava and Rada Mihalcea. 2008. Learning to identify emotions in text. In *Proc. of the 2008 ACM Symposium on Applied Computing (SAC).* ACM Press, 1556–1560.

[33] Robert Tibshirani, Jacob Bien, Jerome Friedman, Trevor Hastie, Noah Simon, Jonathan Taylor, and Ryan J Tibshirani. 2012. Strong rules for discarding predictors in lasso-type problems. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 74, 2 (2012), 245–266.

[34] Sida Wang and Christopher D Manning. 2012. Baselines and bigrams: Simple, good sentiment and topic classification. In *Proc. of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2.* Association for Computational Linguistics, 90–94.

[35] Wenbo Wang, Lu Chen, Krishnaprasad Thirunarayan, and Amit P Sheth. 2012. Harnessing Twitter "Big Data" for Automatic Emotion Identification. In *2012 International Conference on Privacy, Security, Risk and Trust, PASSAT 2012, and 2012 International Conference on Social Computing, SocialCom 2012.* IEEE Computer Society, 587–592.

[36] Svante Wold, Kim Esbensen, and Paul Geladi. 1987. Principal component analysis. *Chemometrics and intelligent laboratory systems* 2, 1-3 (1987), 37–52.