Predicting Emotional Reaction in Social Networks

Jérémie Clos*[1], Anil Bandhakavi*[1], Nirmalie Wiratunga[1], and Guillaume Cabanac[2]

[1] Robert Gordon University
Aberdeen, United Kingdom
`{initial.lastname}@rgu.ac.uk`
[2] Université de Toulouse
Toulouse, France
`{firstname.lastname}@univ-tlse3.fr`

Abstract. Online content has shifted from static and document-oriented to dynamic and discussion-oriented, leading users to spend an increasing amount of time navigating online discussions in order to participate in their social network. Recent work on emotional contagion in social networks has shown that information is not neutral and affects its receiver. In this work, we present an approach to detect the emotional impact of news, using a dataset extracted from the Facebook pages of a major news provider. The results of our approach significantly outperform our selected baselines.

1 Introduction

With the rise of the social web, a majority of online content has shifted from being static and document-oriented to being highly dynamic and discussion-oriented. With this shift, users have been spending more time navigating online discussions in order to stay informed with their social network. Recent work on emotion contagion in social networks [2] suggests that information is not neutral, and the way it is presented has an impact on the emotional state of its consumers. This demonstrates the importance of providing users with a way to control this content. In this work, we present a technique to predict the emotional impact of news on its consumers, using a dataset extracted from the Facebook pages of the New York Times, a major news network.

We highlight the novelty of our work with respect to existing research on textual emotion detection, before formalizing our problem and explaining our methodology. We evaluate our approach using two naive and two strong baselines. We conclude the paper by discussing our positive results and potential extensions of this work.

2 Related work

Our work lies in the broader context of opinion mining. Most of the literature in this area aims to mine either the sentiment (*positive* and *negative*) or the basic emotions (*anger, joy, ...*) expressed in the content using computational models learned from labeled or distantly labeled sentiment or emotion corpora [1, 4, 7]. More recently work has also

*The two authors contributed equally to this work been done on the detection of emotion in a social network, but focusing on analyzing the emotion contained in text rather than its influence on others [5].

The originality of our work lies in predicting emotion reactions induced in readers by emotional text. Whilst harnessing emotion rated content (e. g., news stories) like in [6,8], to learn word-emotion lexicons, we also go a step further and propose methods to adopt such lexicons for predicting emotion reactions towards emotional text (e. g., news posts). The task described in this work is thus inherently harder because of the latent factors that are implied in the process, e. g., a joyful news might be received with anger by a certain population if they already have a negative predisposition towards the entity concerned by the news, and inversely. Analyzing this bias, however, is beyond the scope of this work and is reserved for future research.

3 Method

3.1 Problem definition

We now give a formal outline to the problem of emotion reaction prediction. Given a set of posts P in a social network (e. g., Facebook) and their corresponding emotion rating vectors R, where R_i is the rating vector corresponding to the post P_i, we aim to predict the emotion rating vector r' for an unseen post p'. The emotion ratings for each post in P are normalized to form a probability distribution across the different emotions. For example, a post *friend met with an accident :(* and its emotion ratings vector $\langle anger : 0.35, joy : 0.0, sadness : 0.55, surprise : 0.15, love : 0.0\rangle$.

3.2 Methods

Our approach contains two different steps. First we learn an emotion lexicon from emotion rated Facebook posts, in order to model the emotion distribution of that particular post. Secondly we train a multi-linear regression (MLR) model using the emotion distribution as predictors. The regression model is used to predict the emotion reaction distribution on unseen posts, thus providing a mapping from the emotional state of the post to the emotional state of the users that are reacting to it.

3.3 Lexicon for emotion reaction detection

In this section we describe our proposed unigram mixture model (UMM) applied to the task of emotion lexicon (EMOLEX) generation. We model real-world emotion data as a mixture of emotion bearing words and emotion-neutral (background) words. For example consider the tweet *going to Paris this Saturday #elated #joyous*, which explicitly connotes emotion *joy*. However, the word *Saturday* is evidently not indicative of *joy*. Further *Paris* could be associated with emotions such as *love*. Therefore our generative model assumes a mixture of two unigram language models to account for such word mixtures in documents. More formally our generative model describes the generation of documents connoting emotion e_t as follows:

$$P(D_{e_t}, Z|\theta_{e_t}) = \prod_{i=1}^{|D_{e_t}|} \prod_{w \in d_i} [(1 - Z_w)\lambda_{e_t} P(w|\theta_{e_t}) \\ +(Z_w)(1 - \lambda_{e_t})P(w|N)]^{c(w,d_i)} \quad (1)$$

where θ_{e_t} is the emotion language model and N is the background language model. λ_{e_t} the mixture parameter, $c(w, d_i)$ the number of times word w occurs in document d_i and Z_w a binary hidden variable which indicates the language model that generated the word w.

We can estimate parameters θ_{e_t} and Z using expectation maximization (EM), which iteratively maximizes the complete data (D_{e_t}, Z) by alternating between two steps: E-step and M-step. The E and M steps in our case are as follows:

E-step:

$$P(Z_w = 0|D_{e_t}, \theta_{e_t}^{(n)}) = \frac{\lambda_{e_t} P(w|\theta_{e_t}^{(n)})}{\lambda_{e_t} P(w|\theta_{e_t}^{(n)}) + (1 - \lambda_{e_t})P(w|N)} \quad (2)$$

M-step:

$$P(w|\theta_{\theta_{e_t}}^{(n+1)}) = \frac{\sum_{i=1}^{|D_{e_t}|} P(Z_w = 0|D_{e_t}, \theta_{e_t}^{(n)})c(w, d_i)}{\sum_{w \in V} \sum_{i=1}^{|D_{e_t}|} P(Z_w = 0|D_{e_t}, \theta_{e_t}^{(n)})c(w, d_i)} \quad (3)$$

where n indicates the EM iteration number. The EM iterations are terminated when an optimal estimate for the emotion language model θ_{e_t} is obtained. EM is used to estimate the parameters of the k mixture models corresponding to the emotions in E. The emotion lexicon $EmoLex$ is learned by using the k emotion language models and the background model N as follows:

$$EmoLex(w_i, \theta_{e_j}) = \frac{P(w_i|\theta_{e_j}^{(n)})}{\sum_{t=1}^{k}[P(w_i|\theta_{e_t}^{(n)})] + P(w_i|N)} \quad (4)$$

$$EmoLex(w_i, N) = \frac{P(w_i|N)}{\sum_{t=1}^{k}[P(w_i|\theta_{e_t}^{(n)})] + P(w_i|N)} \quad (5)$$

where k is the number of emotions in the corpus, and $EmoLex$ is a $|V| \times (k+1)$ matrix, where $|V|$ is the size of the vocabulary V.

3.4 Lexicon-based Regression for Emotion Reaction Detection

In this section we describe the multilinear regression model built using feature vectors extracted using the EMOLEX emotion lexicon. The model is built in two stages. In the first stage EMOLEX is used to extract features to represent a post as a 5-dimensional emotion vector, using a simple average and aggregate approach, meaning that each component of the feature vector is computed as an average of the values of the corresponding component for each term in the post. More formally the feature vector d_{vec} for a post d is extracted using the formulation described in equation 6.

$$d_{vec} = \frac{\sum_{w \in d} EmoLex(w) \times count(w, d)}{|d|} \tag{6}$$

Here $EmoLex(w)$ represents the emotion vector corresponding to the word w, $count(w, d)$ the frequency w in the post d and $|d|$ the length of the post. In the second stage we build five separate MLR models, one for each target emotion. We now describe the MLR model for an arbitrary emotion e_k.

Given a matrix of training vectors $D_{n\times 5}$ = $d^1_{vec}, d^2_{vec}, \dots, d^n_{vec}$, and their corresponding user ratings vector $R_{n\times 1}$ = $r^1_{e_k}, r^2_{e_k}, \dots, r^n_{e_k}$, for emotion e_k, the MLR model is defined in equation 7.

$$R = D \times W + \mathcal{E} \tag{7}$$

In this equation W represents the coefficient matrix, which when multiplied with D becomes the fit of the regression model to the data. $\mathcal{E}$ is the vector that captures the deviation of the model. The objective is to learn the coefficient matrix W, which along with D, $\mathcal{E}$, best estimates (i. e., with a minimal training error) the ratings vector R.

4 Evaluation

Given a set of emotionally charged Facebook posts, we investigate techniques to estimate the emotional reactions towards them, captured in the form of numerical ratings: the number of times people clicked on an emotion emoticon. We leverage a Facebook feature which allows users to react to any item published on a user timeline using an emoticon as shown in figure 1..

Fig. 1: Emotional reactions in Facebook stories

We evaluated our method using a stratified k-fold cross validation with 5 folds and the root mean square error (RMSE) as the performance metric. RMSE is a standard performance metric used when estimating continuous quantities, and is thus suited to our task. It is defined in equation 8 where Y is the vector of observed values, $\hat{Y}$ the vector of predicted values and n the number of instances in the dataset.

$$RMSE(Y, \hat{Y}) = \sqrt{\frac{\sum_{i=0}^{n} (\hat{Y}_i - Y_i)^2}{n}} \tag{8}$$

4.1 Baselines

We use two naive baseline methods based on general corpus statistics (UNIFORM and EMPIRICAL) which do not learn any computational model on the training posts in order to predict the emotion distribution of unobserved posts, as well as two stronger contenders: one based on a simple lexicon with a trivial mapping (EMOLEX) and one based on a linear regression trained on a WORD2VEC embedding (WORD2VEC+MLR).

Corpus statistics	
Number of posts	5367
Average terms/sentence	22.34
EMOLEX coverage	18792
WORD2VEC coverage	16011

Emotion probability distribution		
	Posts	Reaction
Anger	0.192	0.220
Joy	0.155	0.104
Sadness	0.208	0.269
Surprise	0.178	0.100
Love	0.264	0.304

Table 1: Descriptive statistics on the New York Times dataset

1. UNIFORM assumes a completely uniform distribution over the target labels, so that no matter the input the output remains the following:
$$f(d) = \langle 0.2; 0.2; 0.2; 0.2; 0.2 \rangle$$
2. EMPIRICAL assumes that the distribution over the target labels is always the same as the empirical distribution observed in the training data, so that regardless of the input the output remains the following:
$$f(d) = \left\langle \frac{f(e_1)}{\sum_{i=0}^{|e|} f(e_i)}; \frac{f(e_2)}{\sum_{i=0}^{|e|} f(e_i)}; \frac{f(e_3)}{\sum_{i=0}^{|e|} f(e_i)}; \frac{f(e_4)}{\sum_{i=0}^{|e|} f(e_i)}; \frac{f(e_5)}{\sum_{i=0}^{|e|} f(e_i)} \right\rangle$$
where $f(e_i)$ is the frequency of emotion i in the training corpus.
3. EMOLEX simply uses the output of the emotion lexicon used to extract the feature vectors as a direct output.
$$f(d) = \langle \text{EMOLEX}_1(d); \text{EMOLEX}_2(d); \text{EMOLEX}_3(d); \text{EMOLEX}_4(d); \text{EMOLEX}_5(d) \rangle$$
where $\text{EMOLEX}_i(d)$ is the output of the lexicon for emotion i and document d.
4. WORD2VEC+MLR uses word vectors from a WORD2VEC embedding [3], computed on a 400-dimensional embedding with a skipgram-10 model on a Wikipedia corpus, and trains a MLR on it.
$$D\prime = \langle v(t_1); v(t_2); ...; v(t_n) \rangle$$
where $v(t_i)$ is the embedding vector for term i belonging to the document.

4.2 Dataset

We used a dataset crawled from the comments on the Facebook page of the New York Times. As detailed in table 1 emotions are not uniformly distributed in the dataset itself, but the distribution of emotions in the Facebook posts is strongly correlated with the distribution of emotions in the reactions ($R = 0.8814$ on a Pearson test). We also note that the coverage of our emotion lexicon is close from the coverage of the WORD2VEC embedding despite the word embedding being computed on a general purpose resource.

4.3 Results

The results of our experiment, shown in Table 2 averaged over 5 folds show that our approach outperforms all the baselines. We note that while our approach outperforms all of the baselines by a significant margin ($p < 0.05$ on a pairwise two-tailed T-test computed on the 5 folds), the biggest margin remains between approaches that used an emotion mapping and approaches that did not. Hence, there is a correlation between the reactions of the users and the emotions displayed in the Facebook stories themselves, which leads more credence to preexisting works on online emotion contagion [2].

	Method	RMSE
Naive baselines	UNIFORM	0.578
	EMPIRICAL	0.532
Strong baselines	EMOLEX	0.510
	WORD2VEC+MLR	0.531
Approach	**EMOLEX+MLR**	**0.492**

Table 2: Results (lower is better)

5 Conclusion

In this work we demonstrated the validity of our approach to predict the emotional reaction to a specific news item. We showed that the mapping from news item to an emotion space fed into a multilinear regression model outperformed both a direct mapping from the text (using WORD2VEC and a multilinear regression) and an estimation from the text (using the EMOLEX emotion lexicon). This work constitutes a first step towards building a generic model for estimating the emotional impact of news and providing users with a way to avoid being manipulated. Future extensions of this work will focus on diversifying the communication platforms used for spreading emotion-rich content, as well as studying the practical effect of such contagion on users.

References

1. Binali, H., Wu, C., Potdar, V.: Computational approaches for emotion detection in text. In: 4th IEEE Inter. Conf. on Digital Ecosystems and Technologies. pp. 172–177. IEEE (2010)
2. Kramer, A.D., Guillory, J.E., Hancock, J.T.: Experimental evidence of massive-scale emotional contagion through social networks. Proc. of the National Academy of Sciences 111(24) (2014)
3. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in NIPS. pp. 3111–3119 (2013)
4. Pang, B., Lee, L.: Opinion mining and sentiment analysis. Foundations and trends in information retrieval 2(1-2), 1–135 (2008)
5. Qiyao, W., Zhengmin, L., Yuehui, J., Shiduan, C., Tan, Y.: Ulm: A user-level model for emotion prediction in social networks. China Universities of Posts and Telecommunications (2016)
6. Rao, Y., Lei, J., Wenyin, L., Li, Q., Chen, M.: Building emotional dictionary for sentiment analysis of online news. World Wide Web 17(4), 723–742 (2014)
7. Ribeiro, F.N., Araújo, M., Gonçalves, P., Gonçalves, M.A., Benevenuto, F.: Sentibench-a benchmark comparison of state-of-the-practice sentiment analysis methods. EPJ Data Science 5(1), 1–29 (2016)
8. Staiano, J., Guerini, M.: Depechemood: a lexicon for emotion analysis from crowd-annotated news. In: Proc. of the 52nd Annual Meeting of the ACL, pp 427-433 (2014)