Sentiment Detection Using PPM

Victoria Bobicev
Technical University of Moldova
victoria.bobicev@gmail.com

Résumé. Détection de sentiments utilisant PPM
Cet article rend compte de notre travail dans le DEFT 2015, Défi Fouille de Texte. Le sujet de ce défi était la fouille d'opinion, l'analyse des sentiments et la détection de l'émotion dans les tweets écrits en français. La tâche a été résolue par un système qui utilise le PPM (Prédiction par correspondance partielle), algorithme de compression basé sur un modèle n-gram statistique. Nous avons présenté deux points: l'algorithme de PPMC basé sur des caractères avec et sans normalisation. Les résultats des expériences avec l'algorithme PPMC basé sur les caractères étaient meilleurs que pour les expériences avec l'algorithme basé sur les mots. La méthode de normalisation appliquée dans le processus de classification afin de surmonter l'imbalanceness des données n'était pas appropriée dans ces conditions et n'a pas aidé à améliorer les résultats.

Abstract.
Sentiment Detection Using PPM.
This paper reports on our work in the DEFT 2015 French Text Mining Challenge. The topic of this challenge was opinion mining, sentiment analysis and emotion detection in tweets written in French. The task was solved by a system that used the PPM (Prediction by Partial Matching) compression algorithm based on an n-gram statistical model. We submitted two runs; character-based PPMC algorithm with normalization and without. The results in the experiments on character based PPMC algorithm were better than word-based. The normalisation method applied in the process of classification in order to overcome the imbalanceness of the data was not appropriate in this case and did not help in improving the results.

Mots-clés : fouille d'opinion, analyse de sentiments, détection d'émotion, Prédiction par correspondance partielle.
Keywords: Opinion Mining, Sentiment Analysis, Emotion Detection, Prediction by Partial Matching.

1 Introduction

The exponential growth of social media and publicly available user-generated content appealed for diversity of data mining tasks. The initial data mining task was the automatic or semi-automatic analysis of large quantities of data to extract some information about a topic and usually to populate a database with the extracted information for the further use. Lately the extraction and aggregation of factual information was supplemented with sentiment and opinion analysis.

Social web data has become the rich source of information about people's sentiments, opinions, preferences and moods and NLP community make every effort to obtain and analyse this information. The NLP challenges are the demonstrative indicator of these efforts. One of the first sentiment analysis task appeared in SemEval 2007, task 14 "Affective text"[1]. The objective of this task was to annotate the short text (news headlines) with the emotion label using a predefined list of emotions (e.g. joy, fear, surprise), and/or for polarity orientation (positive/negative). The next SemEval contained the task of disambiguation of sentiment ambiguous adjectives[2]. The sentiment ambiguous words are pervasive in many languages, the authors of the task wrote in their task description. They concentrated on Chinese, but suggested to use language-independent disambiguation techniques. Twitter was the source of sentiments in SemEval 2013 in the task 2[3]. The task's organisers aimed to promote the research which would lead to a better understanding of

[1] http://nlp.cs.swarthmore.edu/semeval/tasks/task14/summary.shtml
[2] http://semeval2.fbk.eu/semeval2.php?location=tasks#T3
[3] https://www.cs.york.ac.uk/semeval-2013/task2/

how sentiment is conveyed in tweets and texts. They proposed two sub-tasks: an expression-level task and a message-level task. SemEval 2015 continued with the similar task[4] "Sentiment Analysis in Twitter". This time the organisers proposed five sub-tasks: an expression-level task, a message-level task, a topic-related task, a trend task, and a task on prior polarity of terms. In addition to SemEval tasks various other sentiment analysis challenges were announced online, as for example GESTALT: GErman SenTiment AnaLysis shared Task[5]. It included two different tracks: task 1 on mining political debates, and task 2 on product reviews.

This paper reports participation in DEFT 2015[6] French Text Mining Challenge. We participated in two subtasks, Task 1 "Valence Classification of tweets" and task 2 "Fine-grained classification of the tweets". The paper is organised as follows: the next section introduces some related work; section 3 describes the used methodology; the experiments and their results are presented in section 4 which is followed by the discussion and conclusions.

2 Related Work

Text Data Mining intensively analysed sentiments and opinions that appear in consumer-written product reviews (Bisio et al., 2013), financial blogs and political discussions (Kim, Hovy, 2007). Text analysis of user-written online messages has been demanded by the need for such studies from the one hand and an easy access to the online data from the other (Chmiel et al., 2011), (Dodds et al., 2011).

Twitter is one of the most dynamic social nets with very fast reaction to various events. Recently, sentiment and opinion analysis in Twitter become the hot topic. Overcoming the difficulties of classical NLP analysis of tweets (Bontcheva et al., 2013) various applications of tweet sentiment analysis appear consta ntly (Chmiel et al., 2011), (Derczynski et al., 2013), (Hassan et al., 2013).

Sentiment lexicons are the largely used resources for sentiment analysis. Although there are already numerous lexicons with sentiment information such as SentiWordNet (Baccianella et al., 2010), MPQA (Wilson, 2008), SenticNet (Cambria, Hussain, 2012), DepecheMood (Staiano, Guerini, 2014) and others most of them are English; there is lack of similar resources for other languages.

However, the scarcest sources in the sentiment analysis domain are the annotated corpora. Although there were made some efforts to annotate various types of texts with various types of affective information this is still definitely not enough (Sabou et al., 2014). The early works in this domain were performed manually by the skilled linguists (Wiebe et al., 2005), (Boldrini et al., 2010), but this type of annotation was time and effort consuming. (Balahur and Steinberger 2009) discussed the problem of multiple annotators and inter-annotator agreement. They demonstrated that the elaborated annotation guidelines were necessary to obtain good inter-annotation agreement. They had to go through two iterations of annotation and to re-write the annotation guidelines on the base of the annotation errors made during the first iteration.

The later experiments used pre-annotated by the users corpora, as, for example, customers reviews marked with zero to five stars, or simply "thumbs up – thumbs down" (Turney 2002) or tweets with hashtags and emoticons indicating author's sentiment (Pak, Paroubek, 2010), (Kouloumpis et al., 2011). The other methods of sentiment annotation are Amazon Mechanical Turk (Narr et al., 2012) and games with purpose (Hong et al., 2013). The latter paper emphasized the fact that most sentiment resources have been created for English and describes creation of the language independent platform in the form of a game similar with tetris for online sentiment annotation of Korean words.

Although (Narr et al., 2012) created resources for four European languages: English, German, French and Portuguese using Amazon Mechanical Turk, (Hong et al., 2013) pointed out that such method is not socially attractive and well designed online game integrated with social networks such as Facebook and adapted to mobile devices are more appropriate tools for obtaining sentiment related lexical resources such as annotated corpora and lexicons. They also discussed in the conclusion that sometimes three classes of sentiment (positive, neutral, or negative) are not adequate to accurately capture the sentiment perceived by human judges. A partial solution of the problem is addition of granularity to sentiment classes such as sentiment scores in real numbers or even better, introduction of extra dimensions of sentiments as for example, 'anxiousness', 'anger', and 'inhibition'.

[4] http://alt.qcri.org/semeval2015/task10/
[5] https://sites.google.com/site/iggsasharedtask/
[6] https://deft.limsi.fr/2015/

3 Methodology Description

Detection sentiment and opinions in text can be viewed as a type of classification task. As it was discussed in the previous section such tasks are solved using machine learning methods. We used PPM in the sentiment classification experiments.

Class	Number of extracted tweets	Per cent		Number of lost tweets
positive	2435	31.2%		29
negative	1853	23.7%		41
neutral	3523	45.1%		48
total	7811	100%		118

TABLE 1 : The number and percent of tweets annotated as positive, negative and neutral for the task 1.

Class	Number of extracted tweets	Per cent		Number of lost tweets
information	3523	52.9%		48
opinion	2243	33.7%		32
sentiment	82	1.2%		0
emotion	809	12.2%		17
total	6657	100%		97

TABLE 2 : The number and percent of tweets annotated as : information, opinion, sentiment and emotion for the task 2.1.

3.1 Tasks Description

There are different types of classifications in the sentiment analysis domain. The paper describes the experiments for three classification tasks:

– Valence Classification of tweets. The aim of the task was to classify automatically the tweets depending on the opinion, sentiment or emotion expressed in the text: positive, negative, neutral or mixed, when the message held both positive and negative opinions, sentiments or emotions.

– Fine-grained classification of the tweets. The aim of this task was to assess the performance of textual opinion, sentiment and emotion detection system. It was divided into two sub-tasks:

  – Detection of one of the four proposed generic classes of the information expressed in the tweet. The generic classes proposed in this context were: INFORMATION, OPINION, SENTIMENT and EMOTION.
  – Detection of the specific class of the opinion/sentiment/emotion among 18 classes, as proposed in the uComp[7] project: COLÈRE (anger), PEUR (fear), TRISTESSE (sadness), DÉGOÛT (disgust), ENNUI (boredom), DÉRANGEMENT (disturbance), DÉPLAISIR (displeasure), SURPRISE NÉGATIVE (negative surprise),

[7] http://www.ucomp.eu/

APAISEMENT (appeasement), AMOUR (love), PLAISIR (pleasure), SURPRISE POSITIVE (positive surprise), INSATISFACTION (dissatisfaction), SATISFACTION (satisfaction), ACCORD (agreement), VALORISATION (valorization), DÉSACCORD (disagreement) and DÉVALORISATION (devalorization).

Class	Number of extracted tweets	Per cent		Number of lost tweets
valorisation	1487	47.4%		17
devalorisation	393	12.5%		8
peur	269	8.6%		5
desaccord	212	6.8%		4
colere	205	6.5%		5
mepris	173	5.5%		3
accord	151	4.8%		3
satisfaction	73	2.3%		0
deplaisir	47	1.5%		0
tristesse	34	1.1%		2
plaisir	34	1.1%		1
derangement	12	0.4%		1
surprise_negative	10	0.3%		0
apaisement	9	0.3%		0
insatisfaction	9	0.3%		0
amour	8	0.3%		0
ennui	4	0.1%		0
surprise_positive	4	0.1%		0
total	3134	100%		49

TABLE 3 : The number and percent of tweets annotated with 18 classes of sentiments for the task 2.2.

3.2 The Data Description

A set of annotated French tweets provided by DEFT 2015 organisers was used in the experiments. In agreement with Twitter access and usage policy, the organisers only provided tweet identifiers and a toolkit to collect the data from Twitter. Total of 7929 tweets id was provided by the organisers but only 7811 tweets were extracted due to the fact that some authors deleted their tweets after their extraction for the annotation. The class distribution for the first task for the extracted tweets is presented in the table 1.

While for the task 1 annotation was provided for all 7929 tweet id, only 6754 id were annotated for the task 2.1. After extraction 6657 tweets were obtained for the task 2.1. Table 2 presents the statistics for the tweet annotation for this task.

The class information from the task 2.1 was not included in the task 2.2. Thus for the task 2.2 only 3183 tweet id with annotation was provided. We collected 3134 annotated tweets fir this task. The distribution of sentiment annotation for these tweets is reflected in the table 3.

3.3 The Algorithm Description

In this paper, the application of the PPM (Prediction by Partial Matching) model for automatic text classification is explored. Prediction by partial matching (PPM) is an adaptive finite-context method for text compression that is a back-off smoothing technique for finite-order Markov models (Bratko et al., 2006). It obtains all information from the original data, without feature engineering, it is easy to implement and relatively fast. PPM produces a language model and can be used in a probabilistic text classifier.

PPM is based on conditional probabilities of the upcoming symbol given several previous symbols (Cleary and Witten, 1984). The PPM technique uses character context models to build an overall probability distribution for predicting upcoming characters in the text. A blending strategy for combining context predictions is to assign a weight to each context model, and then calculate the weighted sum of the probabilities:

$$P(x) = \sum_{i=1}^{m} \lambda_i \, p_i(x), \qquad (1)$$

where

λ_i and p_i are weights and probabilities assigned to each order *i (i=1…m)*.

For example, the probability of character '***m***' in context of the word '***algorithm***' is calculated as a sum of conditional probabilities dependent on different context lengths up to the limited maximal length:

$$P_{PPM}(\text{'}\boldsymbol{m}\text{'}) = \lambda_5 \cdot P(\text{'}\boldsymbol{m}\text{'} \mid \text{'}\boldsymbol{orith}\text{'}) + \lambda_4 \cdot P(\text{'}\boldsymbol{m}\text{'} \mid \text{'}\boldsymbol{rith}\text{'}) + \lambda_3 \cdot P(\text{'}\boldsymbol{m}\text{'} \mid \text{'}\boldsymbol{ith}\text{'}) +$$
$$+ \lambda_2 \cdot P(\text{'}\boldsymbol{m}\text{'} \mid \text{'}\boldsymbol{th}\text{'}) + \lambda_1 \cdot P(\text{'}\boldsymbol{m}\text{'} \mid \text{'}\boldsymbol{h}\text{'}) + + \lambda_0 \cdot P(\text{'}\boldsymbol{m}\text{'}) + \lambda_{-1} \cdot P(\text{'esc'}), \qquad (2)$$

where

λ_i (i = 1…5) is the normalization weight;

5 is the maximal length of the context;

P('esc') is so called 'escape' probability, the probability of an unknown character.

PPM is a special case of the general blending strategy. The *PPM* models use an escape mechanism to combine the predictions of all character contexts of length *m*, where *m* is the maximum model order; the order 0 model predicts symbols based on their unconditioned probabilities, the default order -1 model ensures that a finite probability (however small) is assigned to all possible symbols. The *PPM* escape mechanism is more practical to implement than weighted blending. There are several versions of the *PPM* algorithm depending on the way the escape probability is estimated. In our implementation, we used the escape method C (Bell et al., 1989), named PPMC. Treating a text as a string of characters, a character-based *PPM* avoids defining word boundaries; it deals with different types of documents in a uniform way. It can work with texts in any language and be applied to diverse types of classification; more details can be found in (Bobicev, 2007). Our utility function for text classification was cross-entropy of the test document:

$$H_d{}^m = -\sum_{i=1}^{n} p^m(x_i) \log p^m(x_i), \qquad (3)$$

where

n is the number of symbols in a text d,

$H_d{}^m$ – entropy of the text d obtained by model m,

$p^m(x_i)$ is a probability of a symbol x_i in the text *d*.

$H_d{}^m$ was estimated by the modelling part of the compression algorithm.

Usually, the cross-entropy is greater than the entropy, because the probabilities of symbols in diverse texts are different. The cross-entropy can be used as a measure for document similarity; the lower cross-entropy for two texts is, the more similar they are. Hence, if several statistical models had been created using documents that belong to different classes and cross-entropies are calculated for an unknown text on the basis of each model, the lowest value of cross-entropy will indicate the class of the unknown text. In this way cross-entropy is used for text classification.

On the training step, we created *PPM* models for each class of documents; on the testing step, we evaluated cross-entropy of previously unseen texts using models for each class. Thus, cross-entropy was used as similarity metrics; the lowest value of cross-entropy indicated the class of the unknown text.

The maximal length of a context equal to 5 in PPM model was proven to be optimal for text compression (Teahan, 1998). In all our experiments with character-based PPM model we used maximal length of a context equal to 5; thus our method is PPMC5.

The character-based *PPM* models were used for spam detection, source-based text classification and classification of multi-modal data streams that included texts. In (Bratko et al., 2006), the character-based PPM models were used for spam detection. In (Bobicev, 2007), the PPM algorithm was applied to text categorization in two ways: on the basis of characters and on the basis of words.

In (Teahan et al., 2000), a PPM-based text model and minimum cross-entropy as a text classifier were used for various tasks; one of them was an author detection task for the well known Federalist Papers[8]. In (Bobicev, Sokolova, 2008), the PPM algorithm was applied to the short text categorization. Character-based model performed almost as well as SVM, the best method among several machine learning methods compared in (Debole, Sebastiani 2004) for the Reuters-21578[9] corpus.

Usually, PPM models are character-based. However, word-based models were also used for various purposes. For example, if texts are classified by the contents, they are better characterized by words and word combinations than by fragments consisting of five letters. For some tasks words can be more indicative text features than character sequences. That's why we decided to use both character-based and word-based models for PPM text classification. In the case of word-based PPM, the context is only one word and an example for formula (1) looks like the following:

$$P_{PPM}(\text{ word}_i) = \lambda_1 \cdot P(\text{ word}_i \mid \text{word}_{i-1}) + \lambda_0 \cdot P(\text{ word}_i) + \lambda_{-1} \cdot P(\text{'esc'}), \qquad (4)$$

where

word_i is the current word;

word_{i-1} is the previous word.

This model is coded as PPMC1 because of the same C escape method and one length context used for probability estimation.

8 The Federalist Papers by Alexander Hamilton, James Madison, John Jay, Digireads Publishing, Neeland Media LLC, 2006.

9 http://www.daviddlewis.com/resources/testcollections/reuters21578/

Training and testing data is distributed quite unevenly in many tasks, for example, in Reuters-21578 corpus. This imbalance drastically affected the results of the classification experiments; the classification was biased towards classes with a larger volume of data for training. Such imbalance class distribution problems were mentioned in (Bobicev, Sokolova, 2008), (Stamatatos, 2009), (Narayanan et al., 2012). Considering the fact that imbalanced data affected classification results in such a substantial way we used a normalization procedure for balancing entropies of the statistical data models.

The first step of our algorithm was training. In the process of training, statistical models for each class of texts were created. This meant that probabilities of text elements were estimated. The next step after training was calculation of entropies of test documents on the basis of each class model. We obtained a matrix of entropies of class statistical models x test documents'. The columns were entropies for the class statistical models and rows were entropies for a given test documents. After this step the normalization procedure was applied. The procedure consisted of several steps:
(1) Mean entropy for each class of texts was calculated on the base of the matrix;
(2) Each value in the matrix was divided by the mean entropy for this class. Thereby we obtained more balanced values and classification improved considerably.

Although the application of PPM model to the document classification is not new, PPM was never applied to the task of sentiment analysis.

In order to evaluate the PPM classification method for sentiment analysis in French tweets a number of experiments were performed. The aim of the experiments was twofold:
- to evaluate the quality of PPM-based sentiment classification;
- to compare letter-based and word-based PPM classification.

4 The Experiments

The experiments were carried out during the DEFT 2015 shared task event. The first set of the experiments was performed on the base of training data released by the organisers in February. The second set consisted of evaluation runs on test data released in May and the results for these experiments were provided by the organizers.

4.1 The First Set of the Experiments

The first set of the experiments consisted in solving task 1, task 2.1 and task 2.2 of the DEFT challenge using PPM classification algorithm. We used two modification of the algorithm: on the base of characters and on the base of words. Taking into consideration the imbalanced class distribution we used normalization procedure. 10-fold cross-validation was used in order to evaluate the performance of the method in case of the task 1 and task 2.1. We used 4 fold cross-validation for the task 2.2 as some of the 18 classes were presented only with 4 tweets (see table 3). Thus, for these classes 3 files were used for training and 1 for test in each run. The results for the task 1 are reflected in the table 4.

Method	**Precision**	**Recall**	**Macroaverage F-score**
Character-based PPMC5 method without normalization	0.58	0.56	0.57
Character-based PPMC5 method with normalization	0.56	0.58	0.56
Word-based PPMC1 method without normalization	0.50	0.52	0.51
Word-based PPMC1 method with normalization	0.50	0.52	0.51

TABLE 4 : The results obtained on character-based and letter-based PPM models with and without normalization for the task 1.

Method	**Precision**	**Recall**	**Macroaverage F-score**
Character-based PPMC5 method without normalization	0.47	0.40	0.43
Character-based PPMC5 method with normalization	0.42	0.42	0.42
Word-based PPMC1 method without normalization	0.47	0.36	0.41
Word-based PPMC1 method with normalization	0.39	0.43	0.41

TABLE 5: The results obtained on character-based and letter-based PPM models with and without normalization for the task 2.1.

Method	**Precision**	**Recall**	**Macroaverage F-score**
Character-based PPMC5 method without normalization	0.23	0.16	0.18
Character-based PPMC5 method with normalization	0.16	0.20	0.18
Word-based PPMC1 method without normalization	0.26	0.14	0.17
Word-based PPMC1 method with normalization	0.13	0.16	0.14

TABLE 6: The results obtained on character-based and letter-based PPM models with and without normalization for the task 2.2.

As it is seen from the tables, the overall results are not very high which indicate that PPM method is not suitable for the sentiment analysis task. We expected word-based method to perform better as it works with words, the units which sentiments were represented in. However this presupposition was also wrong. The character-based method gave better results in all experiments. The possible reason could be that word-based method was loosing all special characters (such as emoticons) which were registered and used by character-based method. It should be noted that normalization did not improve the results as it was expected. It even made them worse for word-based method. In previous cases it helped to improve the results (Bobicev et al., 2013).

4.2 The Second Set of the Experiments

The second set consisted of evaluation runs on test data released in May and the results for these experiments were provided by the organisers. Taking into consideration that word-based method was worse for all tasks and we were allowed to submit no more than 3 experiment runs for each task we decided to submit only two runs of character based method (with normalization and without it) for each task. Thus, we submitted six runs, two runs for task1, two runs for task 2.1 and two runs for task 2.2. The organisers were interested in Precision, thus only this metric was reported. Tables 7, 8 and 9 contain the results reported by the organisers for the task 1, 2.1 and 2.2.

Method	**Micro precision**	**Macro precision**
Character-based PPMC5 method without normalization	0.568	0.558
Character-based PPMC5 method with normalization	0.542	0.547

TABLE 7: The results obtained on character-based PPM model with and without normalization for the task 1.

Method	**Micro precision**	**Macro precision**
Character-based PPMC5 method without normalization	0.495	0.383
Character-based PPMC5 method with normalization	0.376	0.382

TABLE 8: The results obtained on character-based PPM model with and without normalization for the task 2.1.

Method	**Micro precision**	**Macro precision**
Character-based PPMC5 method without normalization	0.478	0.226
Character-based PPMC5 method with normalization	0.289	0.175

TABLE 9: The results obtained on character-based PPM model with and without normalization for the task 2.2.

It is seen from the tables that the results are similar with the results for the first set of the experiments. The normalisation did not help although the data was imbalanced, especially in the task 2.2 where the results were the worse.

5 Discussion and Conclusions

The paper reports on our work in the DEFT 2015 French Text Mining Challenge. Three tasks of tweet sentiment analysis were proposed in the framework of this challenge. All three tasks analysed French tweets about politics and elections in France. We participated in task1: "Valence Classification of tweets" in which the tweets were classified in three classes: (1) positive, (2) negative, (3) neutral and mixed. We also participated in task 2: "Fine-grained classification of the tweets" which consisted of two subtasks. Task 2.1: "Detection of the generic class of the information expressed in the tweet" classified tweets in four classes: information, opinion, sentiment and emotion. Task 2.2: "Detection of the specific class of the opinion/sentiment/emotion" aimed at detecting the class of the opinion, sentiment or emotion among 18 classes.

We used the system that used the PPM (Prediction by Partial Matching) compression algorithm based on character n-gram statistical model for all tasks. We submitted two runs; character-based PPMC algorithm with normalization and without for each of the subtask. We supposed that word-based algorithm would be better in sentiment detection but the experiments demonstrated that this presupposition was wrong. The results of the experiments on character based PPMC algorithm were better than the results of the experiments on word-based PPMC algorithm for all experiments.

The data released for the tasks was very imbalanced as it is seen in the tables 1, 2, and 3. Such situation is quite common in real classification tasks. Working with imbalanced data we developed a normalisation procedure described in the paper but our normalisation method applied in the process of classification in order to overcome the imbalanceness of the data was not appropriate in this case and did not help in improving the results.

References

BACCIANELLA S., ESULI A., SEBASTIANI F. (2010). SentiWordNet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining. *Proceedings of the 7th Conference on International Language Resources and Evaluation*, 2200-2204.

BALAHUR, A. STEINBERGER R (009) Rethinking Sentiment Analysis in the News: from Theory to Practice and back. *Proceedings of the 1st Workshop on Opionion Mining and Sentiment Analysis*.

BELL, T.C., WITTEN, I.H., CLEARY, J. G. (1989). Modeling for text compression. *Computing Surveys*, 21(4), pp. 557-591.

BISIO F., GASTALDO P., PERETTI C., ZUNINO R., CAMBRIA E.(2013) Data intensive review mining for sentiment classification across heterogeneous domains. *ASONAM 2013*.

BOBICEV V. (2007). Comparison of Word-based and Letter-based Text Classification. *Recent Advances in Natural Language Processing V*, pp. 76–80.

BOBICEV V., SOKOLOVA M. (2008). An effective and robust method for short text classification. *Proceedings of the 23rd national conference on Artificial intelligence* - Volume 3, pp. 1444–1445.

BOBICEV V., SOKOLOVA M., EMAM K., JAFER Y., DEWAR B., JONKER E., MATWIN S. (2013). Can Anonymous Posters on Medical Forums be Reidentified? *Journal of Medical Internet Research,* 15(10):e215.

BOLDRINI E., BALAHUR A., MARTÍNEZ-BARCO P., MONTOYO A. (2010). EmotiBlog: a fine-grained annotation schema for labelling subjectivity in the new-textual genres born with the Web 2.0. *Procesamiento del Lenguaje Natural* 45.

BONTCHEVA K., DERCZYNSKI L., FUNK A., GREENWOOD M.A., MAYNARD D., ASWANI N. (2013). TwitIE: An Open-Source Information Extraction Pipeline for Microblog Text. *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2013)*.

BRATKO A., CORMACK G. V., FILIPIC B., LYNAM T. R., ZUPAN B. (2006). Spam filtering using statistical data compression models, *Journal of Machine Learning Research* 7:2673–2698.

CAMBRIA, E., HUSSAIN A. (2012). *Sentic Computing: Techniques, Tools, and Applications*. Springer.

CHMIEL A., SIENKIEWICZ J., THELWALL M., PALTOGLOU G., BUCKLEY K., KAPPAS A., HOLYST J. (2011). Collective Emotions Online and Their Influence on Community. *Life PLoS one*.

CLEARY J., WITTEN I. (1984). Data compression using adaptive coding and partial string matching, *IEEE Trans. Commun.* 32(4):396–402.

DEBOLE F., SEBASTIANI F. (2004). An Analysis of the Relative Hardness of Reuters-21578 Subsets. *Journal of the American Society For Information Science And Technology* 56(6):971–974. DOI: 10.1002/asi.v56:6.

DERCZYNSKI L., YANG B., JENSEN C.S. (2013). Towards Context-Aware Search and Analysis on Social Media Data. *In Proceedings of the Extending Database Technology conference (EDBT 2013)*.

DODDS, P., HARRIS K., KLOUMANN I., BLISS C., DANFORTH C. (2011). Temporal Patterns of Happiness and Information in a Global Social Network: Hedonometrics and Twitter. *PLoS ONE,* 6, e26752.

HASSAN S., FERNANDEZ M., HE Y., ALANI H. (2013). Evaluation Datasets for Twitter Sentiment Analysis. A survey and a new dataset, *the STS-Gold. first ESSEM workshop*.

HONG Y., KWAK H., BAEK Y., MOON S. (2013). Tower of Babel: A Crowdsourcing Game Building Sentiment Lexicons for Resource-scarce Languages. *WWW '13 Companion Proceedings of the 22nd international conference on World Wide Web companion*. Pages 549-556.

KIM, S.-M., HOVY E. (2007). Crystal: Analyzing predictive opinions on the web. *EMNLP-CoNLL*.

KOULOUMPIS E., WILSON T., MOORE J. D. (2011). Twitter Sentiment Analysis: The Good the Bad and the OMG!. *Proceedings of the Fifth International Conference on Weblogs and Social Media*. AAAI Press, p. 538-541.

NARAYANAN A., PASKOV H., GONG N. Z., BETHENCOURT J., STEFANOV E., SHIN E. C. R., AND SONG D. (2012). On the Feasibility of Internet-Scale Author Identification, *in 2012 IEEE Symposium on Security and Privacy (SP)*, pp. 300 – 314.

NARR S., HULFENHAUS M., ALBAYRAK S. (2012). Language-Independent Twitter Sentiment Analysis. *Knowledge Discovery and Machine Learning (KDML)*, LWA, 12-14.

PAK A., PAROUBEK P. (2010). Twitter as a Corpus for Sentiment Analysis and Opinion Mining. *LREC 2010.*

SABOU M., BONTCHEVA K., DERCZYNSKI L., SCHARL A. (2014). Corpus Annotation through Crowdsourcing: Towards Best Practice Guidelines. *9th Language Resources and Evaluation Conference (LREC-2014)*.

STAIANO J., GUERINI M. (2014). DepecheMood: a Lexicon for Emotion Analysis from Crowd-Annotated News. *Proceedings of ACL-2014*.

STAMATATOS E. (2009). A survey of modern authorship attribution methods, *J. Am. Soc. Inf. Sci. Technol.*, vol. 60, no. 3, pp. 538–556.

TEAHAN W. (1998). Modelling English text. *PhD Thesis,* University of Waikato, New Zealand.

TEAHAN W. J., MCNAB R., WEN Y., WITTEN I. H. (2000). A compression-based algorithm for Chinese word segmentation, *Comput. Linguist.*, vol. 26, no. 3, pp. 375–393.

TURNEY P. D. (2002). Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. *ACL '02 Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. Pages 417-424.

WIEBE J., WILSON T., CARDIE C. (2005). Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation*, volume 39, issue 2-3, pp. 165-210.

WILSON T. (2008). Fine-Grained Subjectivity Analysis. PhD Dissertation, *Intelligent Systems Program, University of Pittsburgh*.