Mohamed Medhat Gaber Mihaela Cocea Stephan Weibelzahl
Ernestina Menasalvas Cyril Labbe (Eds.)

SDAD 2012

The 1st International Workshop on Sentiment Discovery from Affective Data

Workshop co-located with The European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)

Bristol, UK, September 28, 2012

Proceedings

Editors' addresses:
School of Computing
University of Portsmouth
Buckingham Building, Lion Terrace
Portsmouth, PO1 3HE
United Kingdom

{mohamed.gaber | mihaela.cocea}@port.ac.uk

Organizing Committee

Mohamed Medhat Gaber, University of Portsmouth, UK
Mihaela Cocea, University of Portsmouth, UK
Stephan Weibelzahl, National College of Ireland
Ernestina Menasalvas, Universidad Politécnica de Madrid, Spain
Cyril Labbe, Université Joseph Fourier, France

Program Committee

Aladdin Ayesh, De Montfort University, UK
Ryan S.J.d. Baker, Worcester Polytechnic Institute, USA
Albert Bifet, University of Waikato, New Zealand
Samhaa El-Beltagy, Nile University, Egypt
Joao Gomes, Institute for Infocomm Research, Singapore
Hany Hassan, Microsoft Corp., USA
Marwan Hassani, RWTH Aachen, Germany
Aboul Ella Hassanien, Cairo University, Egypt
Arnon Hershkovitz, Worcester Polytechnic Institute, USA
Manolis Mavrikis, London Knowledge Lab, Institute of Education, UK
Olfa Nasraoui, University of Louisville, USA
Cristobal Romero Morales, University of Cordoba, Spain
Sherif Sakr, University of New South Wales, Australia
Yanchang Zhao, RDataMining.com, Australia
Indre Zliobaite, Bournemouth University, UK

Contents

New Features for Sentiment Analysis: Do Sentences Matter?

Gizem Gezici[1], Berrin Yanikoglu[1], Dilek Tapucu[1,2], and Yücel Saygın[1]

[1] Faculty of Engineering and Natural Sciences, Sabancı University, Istanbul, Turkey
{gizemgezici,berrin,dilektapucu,ysaygin}@sabanciuniv.edu
[2] Dept. of Computer Engineering, Izmir Institute of Technology, Izmir, Turkey

Abstract. In this work, we propose and evaluate new features to be used in a word polarity based approach to sentiment classification. In particular, we analyze sentences as the first step before estimating the overall review polarity. We consider different aspects of sentences, such as length, purity, irrealis content, subjectivity, and position within the opinionated text. This analysis is then used to find sentences that may convey better information about the overall review polarity. The TripAdvisor dataset is used to evaluate the effect of sentence level features on polarity classification. Our initial results indicate a small improvement in classification accuracy when using the newly proposed features. However, the benefit of these features is not limited to improving sentiment classification accuracy since sentence level features can be used for other important tasks such as review summarization.

Keywords: sentiment analysis; sentiment classification; polarity detection; machine learning

1 Introduction

Sentiment analysis aims to extract the opinions indicated in textual data enabling us to understand what people think about specific issues by analyzing large collections of textual data sources such as personal blogs, review sites, and social media. An important part of sentiment analysis boils down to a classification problem, i.e., given an opinionated text, classifying it as positive or negative polarity and Machine Learning techniques have already been adopted to solve this problem.

Two main approaches for sentiment analysis are lexicon-based and supervised methods. The lexicon-based approach calculates the semantic orientation of words in a review by obtaining word polarities from a lexicon such as the SentiWordNet [5]. While the SentiWordNet [5] is a domain-independent lexicon, one can use a domain-specific lexicon whenever available since domain-specific lexicons better indicate the word polarities in that domain (e.g. the word ”small” has a positive connotation in cell phone domain; while it is negative in hotel domain).

Supervised learning approaches use machine learning techniques to establish a model from a large corpus of reviews. The set of sample reviews form the training data from which the model is built. For instance in [16] [21] , researchers use the Naive Bayes algorithm to separate positive reviews from negative ones by learning the probability distributions of the considered features in the two classes. While supervised approaches are typically more successful, collecting a large training data is often a problem.

Word-level polarities provide a simple yet effective method for estimating a review's polarity, however, the gap from word-level polarities to review-level polarity is too big. To bridge this gap, we propose to analyze word-polarities within sentences, as an intermediate step.

The idea of sentence level analysis is not new. Some researchers approached the problem by first finding subjective sentences in a review, with the hope of eliminating irrelevant sentences that would generate noise in terms of polarity estimation [13], [24]. Yet another approach is to exploit the structure in sentences, rather than seeing a review as a bag of words [8][11][15]. For instance in [8], conjunctions were analyzed to obtain the polarities of the words that are connected with the conjunct. In [9],[14] researchers focused on sentence polarities separately, again to obtain sentence polarities more correctly, with the goal of improving review polarity in turn. The first line polarity has also been used as a feature by [24].

Similar to [24], this work is motivated by our observation that the first and last lines of a review are often very indicative of the review polarity. Starting from this simple observation, we formulated more sophisticated features for sentence level sentiment analysis. In order to do that, we performed an in-depth analysis of different sentence types. For instance, in addition to subjective sentences, we defined pure, short, and no irrealis sentences.

We performed a preliminary evaluation using the TripAdvisor dataset to see the effect of sentence level features on polarity classification. Throughout the evaluation, we observed a small improvement in classification accuracy due to the newly proposed features. Our initial results showed that the sentences do matter and they need to be explored in larger and more diverse datasets such as blogs. Moreover, the benefit of these features is not limited to improving sentiment classification accuracy. In fact, sentence level features can be used to identify the essential sentences in the review which could further be used in review summarization.

Our paper is organized as follows: Section 2 presents our taxonomy of sentiment analysis features, together with the newly proposed features. Section 3 describes the sentence level analysis for defining the features. Section 4 describes the tools and methodology for sentiment classification together with the experimental results and error analysis. Finally, in Section 5 we draw some conclusions and propose future extension of this work.

2 Taxonomy and Formulation of the New Features

We define an extensive set of 19 features that can be grouped in four categories: (1) basic features, (2) features based on subjective sentence occurrence statistics, (3) delta-tf-idf weighting of word polarities, and (4) sentence-level features. These features are listed in Table 1 and using the notations given below and some basic definitions provided in Table 2, they are defined formally in Tables 3-7.

Table 1. Summary Feature Descriptions for a Review R

Group Name	Feature	Name
Basic	F_1	Average review polarity
	F_2	Review purity
Occurrence of Subjective Words	F_3	Freq. of subjective words
	F_4	Avg. polarity of subj. words
	F_5	Std. of polarities of subj. words
$\Delta TF * IDF$	F_6	Weighted avg. polarity of subj. words
	F_7	Scores of subj. words
Punctuation	F_8	# of Exclamation marks
	F_9	# of Question marks
Sentence Level	F_{10}	Avg. First Line Polarity
	F_{11}	Avg. Last Line Polarity
	F_{12}	First Line Purity
	F_{13}	Last Line Purity
	F_{14}	Avg. pol. of subj. sentences
	F_{15}	Avg. pol. of pure sentences
	F_{16}	Avg. pol. of non-irrealis sentences
	F_{17}	$\Delta TF * IDF$ weighted polarity of first line
	F_{18}	$\Delta TF * IDF$ scores of subj. words in the first line
	F_{19}	Number of sentences in review

A review R is a sequence of sentences $R = S_1 S_2 S_3 ... S_M$ where M is the number of sentences in R. Each sentence S_i in turn is a sequence of words, such that $S_i = w_{i1} w_{i2} ... w_{iN(i)}$ where $N(i)$ is the number of words in S_i. The review R can also be viewed as a sequence of words $w_1 .. w_T$, where T is the total number of words in the review.

In Table 2, subjective words (SBJ) are defined as all the words in SentiWordNet that has a dominant negative or positive polarity. A word has dominant positive and negative polarity if the sum of its positive and negative polarity values is greater than 0.5 [23]. $SubjW(R)$ is defined as the most frequent subjective words in SBJ (at most 20 of them) that appear in review R. For a sentence $S_i \in R$, the average sentence polarity is used to determine subjectivity of that sentence. If it is above a threshold, we consider the sentence as subjective, forming $subjS(R)$. Similarly, a sentence S_i is pure if its purity is greater than a fixed threshold τ. We experimented with different values of τ and for evaluation we used $\tau = 0.8$. These two sets form the $subS(R)$ and $pure(R)$ sets respectively.

We also looked at the effect of first and last sentences in the review, as well as sentences containing irrealis words. In order to determine irrealis sentences, the existence of the modal verbs 'would', 'could', or 'should' is checked. If one of these modal verbs appear in the sentence then these sentences are labeled as irrealis similar to [17].

Table 2. Basic definitions for a review R

M	the total number of sentences in R
T	the total number of words in R
SBJ	set of known subjective words
$subjW(R)$	set of most frequent subjective words from SBJ, in R (max 20)
$subjS(R)$	set of subjective sentences in R
$pure(R)$	set of pure sentences in R
$nonIr(R)$	set of non-irrealis sentences in R

2.1 Basic Features

For our baseline system, we use the average word polarity and purity defined in Table 3. As mentioned before, these features are commonly used in word polarity based sentiment analysis. In our formulation $pol(w_j)$ denotes the dominant polarity of w_j of R, as obtained from SentiWordNet, and $|pol(w_j)|$ denotes the absolute polarity of w_j.

Table 3. Basic Features for a review R

F_1	Average review polarity	$\frac{1}{T}\sum_{j=1..T} pol(w_j)$
F_2	Review purity	$\frac{\sum_{j=1..T} pol(w_j)}{\sum_{j=1..T} \lvert pol(w_j)\rvert}$

2.2 Frequent Subjective Words

The features in this group are derived through the analysis of subjective words that frequently occur in the review. For instance, the average polarity of the most frequent subjective words (feature F_4) aims to capture the frequent sentiment in the review, without the noise coming from *all* subjective words.

The features were defined before in some previous work [4]; however, to the best of our knowledge, they considered all words, not specifically subjective words.

2.3 *Δtf*idf* Features

We compute the $\Delta tf{*}idf$ scores of the words in SentiWordNet [5] from a training corpus in the given domain, in order to capture domain specificity [12]. For a word w_i, $\Delta tf{*}idf(w_i)$ is defined as $\Delta tf{*}idf(w_i) = tf{*}idf(w_i,+) - tf{*}idf(w_i,-)$.

Table 4. Features Related to Frequency and Subjectivity

F_3	Freq. of subjective words	$\lvert subjW(R)\rvert / \lvert R\rvert$
F_4	Avg. polarity of subj. words	$\frac{1}{\lvert subjW(R)\rvert} \sum_{w_j \in subjW(R)} pol(w_j)$
F_5	Stdev. of polarities of subj. words	$\sqrt{\frac{1}{\lvert subjW(R)\rvert} \sum_{w_j \in subjW(R)} (pol(w_j) - F_4)^2}$

If it is positive, it indicates that a word is more associated with the positive class and vice versa, if negative. We computed these scores on the training set which is balanced in the number of positive and negative reviews.

Then, we sum up the $\Delta tf * idf$ scores of these words (feature F_6). By doing this, our goal is to capture the difference in distribution of these words, among positive and negative reviews. The aim is to obtain context-dependent scores that may replace the polarities coming from SentiWordNet which is a context-independent lexicon [5]. With the help of context-dependent information provided by $\Delta tf * idf$ related features, we expect to better differentiate the positive reviews from negative ones.

We also tried another feature by combining the two information, where we weighted the polarities of all words in the review by their $\Delta tf * idf$ scores (feature F_7).

Table 5. *Δtf*idf* Features

F_6	$\Delta tf * idf$ scores of all words	$\frac{1}{T} \sum_{j=1..T} \Delta tf * idf(w_j)$
F_7	Weight. avg. pol. of all words	$\frac{1}{T} \sum_{j=1..T} \Delta tf * idf(w_j) \times pol(w_j)$

2.4 Punctuation Features

We have two features related to punctuation. These two features were suggested in [4] and since we have seen that they could be useful for some cases we included them in our sentiment classification system.

Table 6. Punctuation Features

F_8	Number of exclamation marks in the review
F_9	Number of question marks in the review

2.5 Sentence Level Features

Sentence level features are extracted from some specific types of sentences that are identified through a sentence level analysis of the corpus. For instance the first and last lines polarity/purity are features that depend on sentence position; while average polarity of words in subjective/pure etc. sentences are new features that consider only subjective or pure sentences respectively.

Table 7. Sentence-Level Features for a review R

F_{10}	Avg. First Line Polarity	$\frac{1}{N(1)} \sum_{j=1..N(1)} pol(w_{1j})$
F_{11}	Avg. Last Line Polarity	$\frac{1}{N(M)} \sum_{j=1..N(M)} pol(w_{Mj})$
F_{12}	First Line Purity	$\frac{\sum_{j=1..N(1)} pol(w_{1j})}{\sum_{j=1..N(1)} \lvert pol(w_{1j}) \rvert}$
F_{13}	Last Line Purity	$\frac{\sum_{j=1..N(M)} pol(w_{Mj})}{\sum_{j=1..N(M)} \lvert pol(w_{Mj}) \rvert}$
F_{14}	Avg. pol. of subj. sentences	$\frac{1}{\lvert subj(R) \rvert} \sum_{w_j \in subjW(R)} pol(w_j)$
F_{15}	Avg. pol. of pure sentences	$\frac{1}{\lvert pure(R) \rvert} \sum_{w_j \in pure(R)} pol(w_j)$
F_{16}	Avg. pol. of non-irrealis sentences	$\frac{1}{\lvert nonIr(R) \rvert} \sum_{w_j \in nonIr(R)} pol(w_j)$
F_{17}	*Δtf*idf* weighted polarity of 1st line	$\sum_{j=1..T} \Delta tf * idf(w_{1j}) \times pol(w_{1j})$
F_{18}	*Δtf*idf* Scores of 1st line	$\sum_{j=1..T} \Delta tf * idf(w_j)$
F_{19}	Number of sentences in review	M

3 Sentence Level Analysis for Review Polarity Detection

We tried three different approaches in obtaining the review polarity. In the first approach, each review is pruned to keep only the sentences that are possibly more useful for sentiment analysis. For pruning, thresholds were set separately for each sentence level feature. Sentences with length of at most 12 words are accepted as short and sentences with absolute purity of at least 0.8 are defined as pure sentences. For subjectivity of the sentences, we adopted the same idea that was mentioned in [23] and applied it on not words, but sentences in this case.

Pruning sentences in this way resulted in lower accuracy in general, due to loss of information. Thus, in the second approach, the polarities in special sentences (pure, subjective, short or no irrealis) were given higher weights while computing the average word polarity. In effect, other sentences were given lower weight, rather than the more severe pruning.

In the final approach that gave the best results, we used the information extracted from sentence level analysis as features used for training our system.

We believe that our main contribution is the introduction and evaluation of sentence-level features; yet other than these, some well-known and commonly used features are integrated to our system, as explained in the next section.

Our approach depends on the existence of a sentiment lexicon that provide information about the semantic orientation of single or multiple terms. Specifically, we use the SentiWordNet [5] where for each term at a specific function, its positive, negative or neutral appraisal strength is indicated (e.g. "good,ADJ, 0.5)

4 Implementation and Experimental Evaluation

In this section, we provide an evaluation of the sentiment analysis features based on word polarities. We use the dominant polarity for each word (the largest polarity among negative, objective or positive categories) obtained from sentiWordNet. We evaluate the newly proposed features and compare their performance to a baseline system. Our baseline system uses two basic features which are the average polarity and purity of the review. These features are previously suggested in [1] and [22] widely used in word polarity-based sentiment analysis. They are defined in Table 3 for completeness. The evaluation procedure we used in our experiments is described in the following subsections.

4.1 Dataset

We evaluated the performance of our system on a sentimental dataset, TripAdvisor that was introduced by [18] and, [19] respectively. The TripAdvisor corpus consists of around 250.000 customer-supplied reviews of 1850 hotels. Each review is associated with a hotel and a star-rating, 1-star (most negative) to 5-star (most positive), chosen by the customer to indicate his evaluation.

We evaluated the performance of our approach on a randomly chosen dataset from TripAdvisor corpus. Our dataset consists of 3000 positive and 3000 negative reviews. After we have chosen 6000 reviews randomly, these reviews were shuffled and split into three groups as train, validation and test sets. Each of these datasets have 1000 positive and 1000 negative reviews.

We computed our features and gave labels to our instances (reviews) according to the customer-given ratings of reviews. If the rating of a review is bigger than 2 then it is labeled as positive, and otherwise as negative. These intermediate files were generated with a Java code on Eclipse and given to WEKA [20] for binary classification.

4.2 Sentiment Classification

Initially, we tried several classifiers that are known to work well for classification purposes. Then, according to their performances we decided to use Support Vector Machines (SVM) and Logistic regression. SVMs are known for being able to handle large feature spaces while simultaneously limiting overfitting, while Logistic Regression is a simple, and commonly used, well-performing classifier. The SVM is trained using a radial basis function kernel as provided by LibSVM [3]. For LibSVM, RBF kernel worked better in comparison to other kernels on our dataset. Afterwards, we performed grid-search on validation dataset for parameter optimization.

4.3 Experimental Results

In order to evaluate our sentiment classification system, we used binary classification with two classifiers, namely SVMs and Logistic Regression. The reviews with star rating bigger than 2 are positive reviews and the rest are negative reviews in our case, since we focused on binary classification of reviews. Apart from this, we also looked at the importance of the features. The importance of the features will be stated with the feature ranking property of WEKA [20] as well as the gradual accuracy increase, as we add a new feature to the existing subset of features.

For these results, we used grid search on validation set. Then, by these optimum parameters, we trained our system on training set and tested it on testing set.

Table 8. The Effects of Feature Subsets on TripAdvisor Dataset

Feature Subset	Accuracy (SVM)	Accuracy (Logistic)
Basic (F1,F2)	79.20%	79.35%
Basic (F1,F2) + $\Delta TF * IDF$ (F6,F7)	80.50%	80.30%
Basic (F1,F2) + $\Delta TF * IDF$ (F6,F7) + ... Freq. of Subj. Words (F3)	80.80%	80.05%
Basic (F1,F2) + $\Delta TF * IDF$ (F6,F7) + ... Freq. of Subj. Words (F3) + Punctuation (F8,F9)	80.20%	79.90%
Basic (F1,F2) + $\Delta TF * IDF$ (F6,F7) + ... Occur. of Subj. Words (F3-F5)	80.15%	79.00%
All Features (F1-F19)	**80.85%**	**81.45%**

Table 9. Comparative Performance of Sentiment Classification System on TripAdvisor Dataset

Previous Work	Dataset	F-measure	Error Rate
Gindl et al (2010) [6]	1800	0.79	-
Bespalov et al (2011) [2]	96000	-	7.37
Peter et al (2011) [10]	103000	0.82	-
Grabner et al (2012) [7]	1000	0.61	-
Our System (2012)	6000	0.81	-

The results for the best performing feature combinations described in Table 1, are given in Table 8. As can be seen in this table, using sentence level features bring improvements over the best results, albeit small.

4.4 Discussion

As can be seen in the experiments section, our system with the newly proposed features obtains one of the best results obtained so far, except for [2]. Although [2] obtains the best result on a large TripAdvisor dataset, its main drawback is that topic models learned by methods such as LDA requires re-training when a new topic comes. In contrast, our system uses word polarities; therefore it is very simple and fast. For this reason, it is more fair to compare our system with similar systems in the literature.

5 Conclusions and Future Work

In this work, we tried to bridge the gap between word-level polarities and review-level polarity through an intermediate step of sentence level analysis of the reviews. We formulated new features for sentence level sentiment analysis by an in-depth analysis of the sentences. We implemented the proposed features and evaluated them on the TripAdvisor dataset to see the effect of sentence level features on polarity classification. We observed that the sentence level features have an effect on sentiment classification, and therefore, we may conclude that sentences do matter in sentiment analysis and they need to be explored for larger and more diverse datasets such as blogs. For future work, we will evaluate each feature set both in isolation and in groups, and work on improving the accuracy. Furthermore, we will switch to a regression problem for estimating the star rating of reviews.

Sentence level features have other uses since they can be exploited further to identify the essential sentences in the review. We plan to incorporate sentence level features for highlighting the important sentences and review summarization in our open source sentiment analysis system SARE which may be accessed through http://ferrari.sabanciuniv.edu/sare.

Acknowledgements. This work was partially funded by European Commission, FP7, under UBIPOL (Ubiquitous Participation Platform for Policy Making) Project (www.ubipol.eu).

References

1. Ahmed, A., Hsinchun, C., Arab, S.: Sentiment analysis in multiple languages: Feature selection for opinion classification in Web forums. ACM Transactions on Information Systems 26, 1–34 (2008)
2. Bespalov, D., Bai, B., Qi, Y., Shokoufandeh, A.: Sentiment classification based on supervised latent n-gram analysis. In: ACM Conference on Information and Knowledge Management (CIKM) (2011)
3. Chang, C.C., Lin, C.J.: Libsvm: a library for support vector machines (2001)
4. Denecke, K.: How to assess customer opinions beyond language barriers? In: ICDIM. pp. 430–435. IEEE (2008)

5. Esuli, A., Sebastiani, F.: Sentiwordnet: A publicly available lexical resource for opinion mining. In: In Proceedings of the 5th Conference on Language Resources and Evaluation (LREC06. pp. 417–422 (2006)
6. Gindl, S., Weichselbraun, A., Scharl, A.: Cross-domain contextualization of sentiment lexicons. Media (2010)
7. Grbner, D., Zanker, M., Fliedl, G., Fuchs, M.: Classification of customer reviews based on sentiment analysis. Social Sciences (2012)
8. Hatzivassiloglou, V., Mckeown, K.R.: Predicting the semantic orientation of adjectives. In: Proceedings of ACL-97, 35th Annual Meeting of the Association for Computational Linguistics. pp. 174–181. Association for Computational Linguistics (1997)
9. Kim, S.m., Hovy, E., Rey, M.: Automatic detection of opinion bearing words and sentences pp. 61–66
10. Lau, R.Y.K., Lai, C.L., Bruza, P.B., Wong, K.F.: Leveraging web 2.0 data for scalable semi-supervised learning of domain-specific sentiment lexicons. In: Proceedings of the 20th ACM international conference on Information and knowledge management. pp. 2457–2460. CIKM '11, ACM, New York, NY, USA (2011)
11. Mao, Y., Lebanon, G.: Isotonic conditional random fields and local sentiment flow. In: Advances in Neural Information Processing Systems (2007)
12. Martineau, J., Finin, T.: Delta tfidf: An improved feature space for sentiment analysis. In: Adar, E., Hurst, M., Finin, T., Glance, N.S., Nicolov, N., Tseng, B.L. (eds.) ICWSM. The AAAI Press (2009)
13. Mcdonald, R., Hannan, K., Neylon, T., Wells, M., Reynar, J.: Structured models for fine-to-coarse sentiment analysis. Computational Linguistics (2007)
14. Meena, A., Prabhakar, T.V.: Sentence level sentiment analysis in the presence of conjuncts using linguistic analysis. Symposium A Quarterly Journal In Modern Foreign Literatures (2), 573–580 (2007)
15. Pang, B., Lee, L.: A sentimental education : Sentiment analysis using subjectivity summarization based on minimum cuts. Cornell University Library (2004)
16. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up? sentiment classification using machine learning techniques. In: Proceedings of EMNLP. pp. 79–86 (2002)
17. Taboada, M., Brooke, J., Tofiloski, M., Voll, K., Stede, M.: Lexicon-based methods for sentiment analysis. Comput. Linguist. 37(2), 267–307
18. The TripAdvisor website. `http://www.tripadvisor.com` (2011), [TripAdvisor LLC]
19. Wang, H., Lu, Y., Zhai, C.: Latent aspect rating analysis on review text data: A rating regression approach. Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining pp. 783–792 (2010)
20. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann (2005)
21. Yu, H.: Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. Proceeding EMNLP 03 Proceedings of the 2003 conference on Empirical methods in natural language processing (2003)
22. Zhai, Z., Liu, B., Xu, H., Jia, P.: Grouping product features using semi-supervised learning with soft-constraints. In: Huang, C.R., Jurafsky, D. (eds.) COLING 2010, 23rd International Conference on Computational Linguistics, Proceedings of the Conference, 23-27 August 2010, Beijing, China. pp. 1272–1280. Tsinghua University Press (2010)
23. Zhang, E., Zhang, Y.: Ucsc on rec 2006 blog opinion mining. In: TREC (2006)

24. Zhao, J., Liu, K., Wang, G.: Adding redundant features for crfs-based sentence sentiment classification. In: Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing. pp. 117–126 (2008)

Product Reputation Model: An Opinion Mining Based Approach

Ahmad Abdel-Hafez, Yue Xu, Dian Tjondronegoro
School of Electrical Engineering and Computer Science
Queensland University of Technology
Brisbane, Australia
ahmad.abdelhafez@student.qut.edu.au, {yue.xu,dian}@ qut.edu.au

Abstract. Product rating systems are very popular on the web, and users are increasingly depending on the overall product ratings provided by websites to make purchase decisions or to compare various products. Currently most of these systems directly depend on users' ratings and aggregate the ratings using simple aggregating methods such as mean or median [1]. In fact, many websites also allow users to express their opinions in the form of textual product reviews. In this paper, we propose a new product reputation model that uses opinion mining techniques in order to extract sentiments about product's features, and then provide a method to generate a more realistic reputation value for every feature of the product and the product itself. We considered the strength of the opinion rather than its orientation only. We do not treat all product features equally when we calculate the overall product reputation, as some features are more important to customers than others, and consequently have more impact on customers buying decisions. Our method provides helpful details about the product features for customers rather than only representing reputation as a number only.

Keywords: reputation model, opinion mining, features impact, opinion strength

1 Introduction

Many websites nowadays provide a rating system for products, which is used by customers to rate available products according to their own experience. Reputation systems provide methods for collecting and aggregating users' ratings to calculate the overall reputation for products, users, or services [2]. This final rate is very important, as it represents the electronic 'word of mouth' that customers build their trust in a product on. On the other hand, most websites allow customers to add textual reviews to explain more about their opinion to the product. These reviews are available for customers to read, to the best of our knowledge, they are not analyzed and counted in the product overall reputation. Many reputation models have been proposed, but most of them concentrated on user's reputation in C2C (Consumer to Consumer) websites such as eBay.com, while service and product reputation has received less attention. Besides, most of the literature about product reputation models neglected users' reviews and counted users' ratings only. Therefore, their reputation systems did not provide any summaries and details about the weakness and strength points in the product.

In this work we will provide a reputation model for products using sentiment analysis methods. The proposed model generates reputation for a specific product depending on the textual reviews provided by users rather than depending on their ratings because users' ratings do not reveal an actual reflection for the products' features, and they do not provide details for customers about features reputation and about "why" the reputation is high or low. For example, a strict user might give three stars for the product although he does not have a clear negative opinion about the product. On the other hand a more generous customer might have a couple of negative opinions about the product but still give four stars. Additionally, textual reviews can be used to provide summaries about product features reputation in addition to the aggregated value for the product reputation, which can make the reputation system more meaningful rather than being just a number. We calculate features impact by counting how many times every feature is mentioned explicitly in the text reviews, assuming that features that are mentioned more by users are more important for them.

In the rest of this paper, we will demonstrate couple of existing product reputation model in the section II, and in the following sections we will explain equations we use to calculate the reputation value for a product. We will also provide diagrams to show the difference between the results of our reputation calculation method and the regular average method used by most websites to represent the overall product reputation.

2 Related Work

2.1 Reputation Models

Reputation models have been studied intensively by many researchers in the last decade, many of these researches concentrated on user's reputation and some of them have discussed product reputations. One of the most basic works on ratings aggregation analyzed robustness of different aggregators, in particular the mean, weighted mean, median and mode, and proposed that using median or mode is more efficient than using mean [1]. Cho et al. [3] proposed a more sophisticated model, they calculated user reputation and used it in order to calculate weights for different ratings. Moreover, they assumed that some users tend to give higher ratings than others, hence, they calculated rating tendency for users and deducted it from user rating. They used the user's accurate prediction and the degree of his activity to define his level of expertise, and then they used this value to represent user's reputation. This method might not be an accurate way to give different weights for ratings, because a user's reputation should not reduce the weight of his opinion about a product. On the other hand, another promising work introduced by Leberknight et al. [4], discussed the volatility of online ratings, where authors aimed to reflect the current trend of users' ratings, they used weighted average where old ratings have less weight than current ones. They introduced a metric called Average Rating Volatility (ARV) that captured the extent of fluctuation present in the ratings, and then they used it to calculate discounting factor, which is used in weighting older ratings.

2.2 Opinion Mining

Many literatures have focused on extracting useful information from the huge amount of available users' opinions in the internet. Opinion mining was used in many different domains. Business Intelligence is the most popular one, where many studies concentrated on mining customers' reviews for better market understanding [5]. Researchers focused on the sentiment analysis part and represented product reputation as a simple count of positive and negative sentiments [6] [7]. Turney [8], Pang et al. [9], and Kamps et al. [10] provided different methods to determine the orientation of a word as positive or negative. In contrast, Hu & Liu [6] proposed a set of techniques for mining and summarizing product reviews to provide a feature based summary of customer reviews, they searched for frequent noun and noun phrases as candidate features. While Popescu et al. [11] identified parts and features of a product depending on finding relation between noun words and the product class using PMI algorithm [8]. Morinaga et al. [12] were one of the first researchers to introduce a general framework for collecting and analyzing users' reviews in order to find the overall product reputation. They used two dimensional positioning Maps, which contained the extracted opinion phrases and associate products with them. The distance between opinion-phrases and products represents closeness. Their proposed method does not mine product features [6], which might be crucial element in the product reputation analysis. In contrast, Hashimoto & Shirota [13] depended on buzz marketing sites to provide a framework for reputation analysis considering product's features. They attempted to discover the topic of each review as initial step, and then they determined important topics depending on the contribution rate of each topic and the polarity of the messages. Finally, the results are visualized for users. However, the effectiveness of their framework has not been evaluated, and the visualization method used to represent the results has not been perfected. Moreover, they neglected topics with lower contributions which might affect the overall product reputation.

To the best of our knowledge none of the previous work has proposed a convenient method to calculate product reputation, depending on the outcome of mining users' reviews. Most of the available methods represent the reputation as a simple count or average of positive and negative opinions in the reviews. While the convenient represented models depended on users' ratings rather than users' textual reviews.

3 The Proposed Approach

3.1 Definition

A product can be described by a set of features representing its characteristics. Some of the features may be more specific or more general than others. For example, for a specific mobile phone product, the "Mobile Camera" is considered as a general feature, while Resolution, Optical Zoom, Flash Light, Video Recording are more specific features of Mobile Camera. In this paper, we define product features as a hierarchy.

Definition 1 (Feature hierarchy): A feature hierarchy consists of a set of features and their relationships, denoted as $FH = \{F, L\}$, F is a set of features where $F = \{f_1, f_2, ..., f_n\}$ and L is a set of relations. In the feature hierarchy, the relationship between a pair of features is the sub-feature relationship. For $f_i, f_j \in F$, if f_j is a sub-feature of f_i, then $(f_i, f_j) \in L$, which means, f_j is more specific than f_i. The root of the hierarchy represents the product itself, and the first level children are the generic features. In this paper, we assume that the feature hierarchy is available.

Definition 2 (User's Review): R is a set of reviews where $R = \{r_1, r_2, ..., r_m\}$. Every review consists of a number of opinions about different features, denoted as $\forall r_i \in R \;\; r_i = \{(f_{i1}, o_{i1}, s_{i1}), ..., (f_{in}, o_{in}, s_{in})\}$. o_{ij} is the orientation of the opinion; $o_{ij} \in \{Pos, Neg, Neu\}$, which represents positive, negative, and neutral respectively. s_i is the strength of the opinion, $s_i \in \{1,2,3\}$, where 1 represents "weak opinion", 2 for "moderate", and 3 for "strong opinion".

In this paper, we assume that the product features and the opinion orientation and strength to the features in each product review have been determined by using existing opinion mining techniques. The proposed reputation model will generate product reputation based on the opinion orientation information, i.e., this information is available, and is the input to the reputation model. There are different methods that can be used to extract this information [14] [15].

For a specific feature f_j, the set of negative reviews are denoted as $R_j^{neg} = \{r_i | o_{ij} = Neg\}$, $R_j^{neg} = \left\{r_1^{neg}, r_2^{neg}, ..., r_{|R_j^{neg}|}^{neg}\right\}$, where $\forall r_i^{neg} \in R_j^{neg}$, $r_i^{neg} = \{(f_{i1}^{neg}, o_{i1}^{neg}, s_{i1}^{neg}), ..., (f_{ij}^{neg}, o_{ij}^{neg}, s_{ij}^{neg}), ...\}$

The same definitions also apply for the set of positive reviews R_j^{pos}. The neutral orientation reflects the lack of opinion about the specific feature and consequently will not be considered in the reputation model.

Our proposed product reputation model consists of three stages:

- Feature Reputation: the reputation of every feature is calculated based on the frequencies of positive and negative opinions about the features and its sub features.
- Features' Impact: feature impact is used to give a different weight for every feature depending on the number of opinions available in users' reviews about this feature.
- Product Reputation: the final product reputation is the aggregation of features' reputations.

In the following sections we will describe them in details.

3.2 Feature Reputation

The basic idea of the proposed model is to generate the reputation of a product based on the reputation of the product's features. The reputation of each feature is generated based on the opinion orientation and strength of its sub features. For a feature f_i, the reputation of f_i will be the aggregation of the positive and negative opinions weights for all of its sub-features f_j, where $(f_i, f_j) \in L$ as mentioned in Definition 1. This section will discuss how to derive feature reputation based on sub features' opinion information.

Negative Opinion Weight

In this part we suggest a formula to give more weights for frequent negative opinions about a specific feature. By "frequent", we mean that the negative opinion about a feature has occurred in many reviews. Frequent negative opinions may indicate a real drawback in the product, where there is a larger probability that a customer will have the same problem if he buys this product. Thus, when more reviews share a negative opinion about the same feature, the risk of facing the same problem becomes higher. These kinds of problems must appear in the reputation model in order to reflect a true evaluation for the product in use, and to draw user's attention so that he can look for more details and have a rational decision about buying the product. Therefore, we suggest giving these types of negative opinions more weight to draw the user's attention to problems in the products. If we have some negative opinions about different sub-features f_j for the feature f_i, we do not consider them as frequent for the feature f_i. For example, if we have negative opinions about a mobile phone camera as follows "Low video recording quality", "The flash light give a very harsh light", and "No zoom available", these negative opinions about the camera cannot be considered frequent in terms of "camera" because they are about different sub-features (video recording, flash light, and zoom) of the generic feature "camera". Equation (1) is used to calculate the negative weights for each feature f_j.

$$N_j = \sum_{i=1}^{\left|R_j^{neg}\right|} \left(s_{ij}^{neg} + \frac{i+\beta-1}{\beta} - 1 \right) \tag{1}$$

N_j: is the weight for negative opinions of feature f_j.

$\left|R_j^{neg}\right|$: is the number of reviews that contains negative opinions about the feature f_j.

s_{ij}^{neg} : is the strength of negative opinion in review (*i*) about the feature (*j*).

β : is a positive integer that is used to define the interval of weight increment for the subsequent opinions, where

$$Interval = \frac{1}{\beta} \tag{2}$$

The value of β is subject to change, higher β values will furnish higher feature reputation values, and that is because the $Interval$ value in (2) will be less, which indicates fewer increments in weights for frequent negative opinions. We use ($\beta = 3$); which indicates that the weights for frequent opinions will match with the series in (3), as it appears in the series we keep the value of the opinion strength s_i intact, and we add $Interval$ to increase the weight.

$$N_j = s_{1j}^{neg} + (s_{2j}^{neg} + 0.33) + \ldots + (s_{|R_j^{neg}|j}^{neg} + \frac{|R_j^{neg}| - 1}{3}) \tag{3}$$

For a feature f_i which has sub features {*f₁*, *f₂*, ..., *f_k*}, Equation (4) is proposed to calculate the overall weight for negative opinions about the generic feature f_i, which is the sum of the weights of all its sub-features calculated using Equation (1).

$$WN_i = \left(\sum_{j=1}^{k} N_j \right) + N_i \tag{4}$$

WN_i: is the weight of all negative opinions about a generic feature f_i in the hierarchy *FH*.

k : is the number of sub-features of feature f_i .

N_i: represents the weight of negative opinions about the generic feature f_i itself and not about one of its sub-features. It is calculated using Equation (1).

Positive Opinion Weight

For the positive opinions, we propose to calculate the positive weight for a feature f_j by adding opinion strength values s_i given in Equation (5). If the feature has sub features $\{f_1, f_2, ..., f_k\}$, the overall weight for positive opinions about the generic feature f_i, is the sum of the positive weights of all its sub-features plus the positive weight of itself, as showed in Equation (6) below:

$$P_j = \sum_{i=1}^{\left|R_j^{pos}\right|} s_{ij}^{pos} \tag{5}$$

$$WP_i = \left(\sum_{j=1}^{k} P_j\right) + P_i \tag{6}$$

P_j: is the weight for positive opinions of feature f_j.

WP_i: is the weight of all positive opinions about a generic feature f_i in the hierarchy *FH*.

P_i: represents the weight of positive opinions about the generic feature f_i itself and not about one of its sub-features. It is calculated using Equation (5).

Calculating Feature Reputation

In this paper, we propose to calculate the reputation of a feature based on its overall positive and negative weights as showed in Equation (7), which represents the percentage of positive opinion weights to the total weights of both positive and negative opinions.

$$FREP_i = \frac{WP_i}{WP_i + WN_i} \times 100 \tag{7}$$

An example is given in Table 1 to demonstrate the proposed method. In the table, for simplicity, each feature listed on the left most column has three sub features; NOf_1, NOf_2, and NOf_3 are the number of reviews which contain negative opinions to the corresponding sub features; N_1, N_2, and N_3 are the negative weight of corresponding sub features; NOF_i and $WNOF_i$ are the number of reviews containing negative opinions about feature F_i and its negative weight respectively. It also shows the total number of positive reviews (*PO*), the total number of negative reviews (*NO*), overall weight for positive (WP_i) and negative (WN_i) opinions, and the aggregation ($FREP_i$) using the proposed method and the (*PPR*) which is the percentage of positive reviews among all reviews without considering the strength of opinion and it can be calculated using Equation (7), where (WP_i=*PO*) and (WN_i=*NO*). (Note: the strength of each opinion was not provided in the table).

The example shows the detailed calculations for both positive and negative opinions weights. In the last two columns we can see the differences between the feature reputation value using our method ($FREP_i$) and the simple average method (*AVG*). Our method results in lower reputation in all cases, this is logical as we give more weight for negative opinions. For example, the total number of negative opinions (*NO*) for both F_2, and F_7 are the same which is equal to 21. Nevertheless, the overall weight for negative opinions (N_2) for F_2 is 63.33 and for F_7 is 53.00, which is totally different. This difference is due to; first, the large frequency for the second subfeature (NOf_2 = 11) for F_2, second, higher values for opinions' strength (which was not provided in the table). Fig. 1 shows the relation between ($FREP_i$) and (*AVG*) where the difference between the two values is the most when the percentage of negative opinions to positive ones is higher. And this complies with our purpose of giving negative opinions more weight.

3.3 Feature Impact

Depending on the fact that product features are not equally important to customers, we will calculate feature's impact, which is a value that reflects a feature's influence between users. Some of the features are essential for a product to work, but they do not inspire customers to buy the product, as they become consistent over time. On the other hand, some hot features, that are improved continually or new features have high influence on customers to be more interested in the product. Thus, these features should have more impact on the product overall reputation. Features impact will be used to give different weights for every feature in the final product reputation aggregation formula. We suggest that features that frequently occurred in users' reviews have more impact than other features. Let *Mj* denote the number of reviews that have opinion about this feature, whether positive or negative, the impact of a feature f_j, denoted as $I_{j,}$ is defined in Equation (8) below:

$$M_j = \left|R_j^{neg}\right| + \left|R_j^{pos}\right|$$

$$I_j = \frac{M_j}{M_{Max}} \tag{8}$$

M_{Max}: is the largest value of M_j for all features.

All feature impacts will be given values between 0 and 1; 1 for the feature that was mentioned the most in the users' reviews, and thus has the most influence on users.

Table 1. An example showing the calculation of feature reputation

Features	***NOf₁***	***N₁***	***NOf₂***	***N₂***	***NOf₃***	***N₃***	***NOFᵢ***	***WNOFᵢ***	***PO***	***WPᵢ***	***NO***	***WNᵢ***	***FREPᵢ***	***PPR***
F_1	2	4.33	4	9	1	3	5	15.33	110	266	12	37.67	87.60	90.16
F_2	3	6	**11**	37.33	3	9	4	11	87	170	**21**	**63.33**	72.86	80.56
F_3	10	39.00	9	26	7	22	0	0	215	425	26	87.00	83.01	89.21
F_4	7	19	6	15	3	7	2	4.33	366	722	18	45.33	94.09	95.31
F_5	**13**	49	8	25.33	2	3.33	1	1	145	283	24	78.67	78.25	85.80
F_6	9	30	11	38.33	5	14.33	17	78.33	417	835	**42**	**161.00**	83.84	90.85
F_7	**8**	20.33	5	14.33	3	5	5	13.33	329	655	**21**	**53.00**	92.51	94.00
F_8	12	47	2	6.33	3	6	0	0	273	563	17	59.33	90.47	94.14

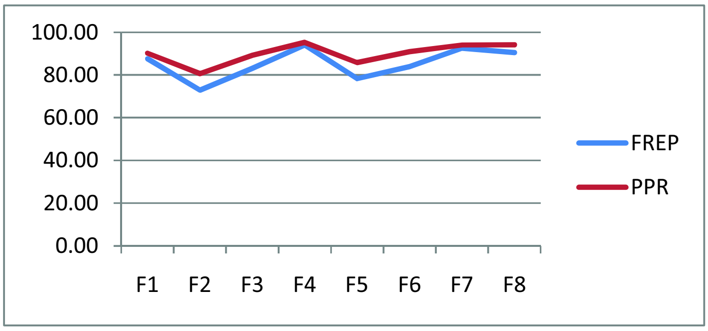


Fig. 1. Feature reputation diagram for the proposed method and the simple average method

3.4 Product Reputation

Many opinions in customers' reviews targeted the product itself rather than mentioning a specific feature in the product, these opinions are also considered in our model. We propose to calculate the product reputation by integrating the reputation calculated based on the reviews which are directly about the product and the reputations of the product's direct features.

Assume that a product has *h* direct sub features, $FREP_j$ and I_j are the reputation and the impact of each sub feature, respectively. Let *GOP* denote the product reputation calculated using Equation (7) where WN_i and WP_i are the number of negative and positive opinions about the product itself in the reviews respectively, and *GOP* have the impact of 1. The following equation is proposed to calculate the product's overall reputation, where every feature reputation, calculated using Equation (7), is multiplied by its impact, calculated using Equation (8), in order to give different weights for features, plus the *GOP*, and the total is divided by the summation of all features' impacts plus 1 that represents the GOP impact.

$$PR = \frac{\sum_{j=1}^{h}(FREP_j \times I_j) + GOP}{\sum_{j=1}^{h} I_j + 1} \tag{9}$$

Table 2 shows the results of calculating the overall product reputation using our model, and the simple average technique. It shows the values of ($FREP_i$) and (*PPR*), from Table 1, the (M_j) column indicates how many times this feature and its subfeatures have been explicitly mentioned in the reviews, and (I_j) column is calculated using (8) where M_{Max} = 459 (the most mentioned feature). It also shows the results of the product reputation (*PR*) and the regular (*AVG*).

Table 2. Example Reputation Calculation

Features	*FREP$_j$*	*PPR*	*M$_j$*	*I$_j$*	*FREP$_i$ * I$_j$*
F$_1$	87.60	90.16	122	0.27	22.90
F$_2$	72.86	80.56	108	0.24	15.71
F$_3$	83.01	89.21	241	0.53	41.05
F$_4$	94.09	95.31	384	0.84	77.06
F$_5$	78.25	85.80	169	0.37	26.09
F$_6$	83.84	90.85	**459**	1.00	77.51
F$_7$	92.51	94.00	350	0.76	68.36
F$_8$	90.47	94.14	290	0.63	55.05
GOP	**86.31**	**86.31**	**528**	**1.00**	**86.31**
Total	**-**	-	-	5.63	470.03
AVG	**-**	**89.59**	-	-	-
PR	**87.00**	-	-	-	-

As we mentioned before, our model reveals a final reputation lower than the average method. One of the strength points in our model is data representation, as we are able to provide details for customers about every specific feature. Fig. 2 shows the reputation of every feature, which can be more inspiring for customers than the one value reputation representation. Furthermore, more detailed information can also be provided as showed in the example in Fig. 3. For example, if the user is interested in a specific feature and he wants to see more about it, a second level will show the details of negative opinions about sub-features and the frequency of each one.

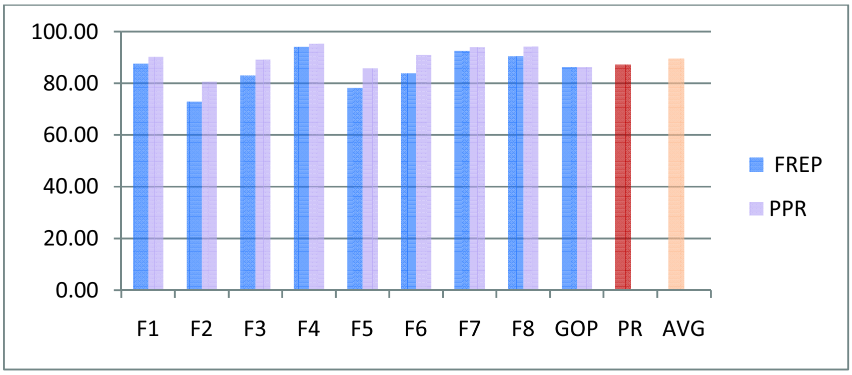


Fig. 2. Results of product reputation model including all features and the regular average result

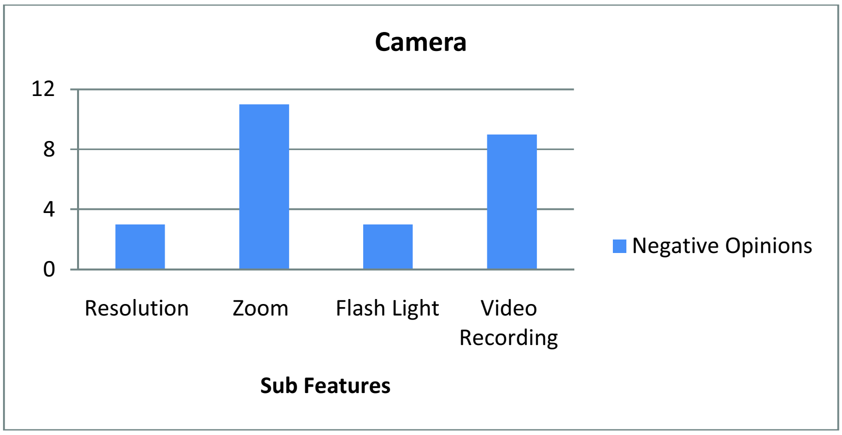


Fig. 3. Example of negative opinions of features at the second level

4 Conclusion

In this paper we have presented a new reputation model for products, our model used text reviews rather than users' ratings. We extracted opinions about hierarchy of features and calculated the frequencies for positive and negative opinions assuming that frequent negative opinions about features and sub-features should get more weight in the reputation calculation, as they indicate a problem in a product a customer may face if they buy it. In Addition, we calculated the impact of features, hence certain features in some products are more inspiring for users, and therefore they are more important in the reputation model. Our model integrates the strength of opinions and provides summary about users' opinions for customers rather than representing reputation as a number of stars. For future work, the reputation model may be modified to consider age and validity of reviews, and also detect malicious users' reviews which aim to sabotage the reputation of a product.

References

1. F. Garcin, B. Faltings, and R. Jurca, "Aggregating reputation feedback," in *International Conference on Reputation ICORE*, 2009, pp. 119-128.
2. P. Resnick, K. Kuwabara, R. Zeckhauser, and E. Friedman, "Reputation systems," *Communications of the ACM (CACM),* vol. 43, pp. 45-48, 2000.
3. J. Cho, K. Kwon, and Y. Park, "Q-rater: A collaborative reputation system based on source credibility theory," *Expert Systems with Applications,* vol. 36, pp. 3751-3760, 2009.
4. S. Leberknight, S. Sen, and M. Chiang, "On the Volatility of Online Ratings: An Empirical Study " in *E-Life: Web-Enabled Convergence of Commerce, Work, and Social Life*. vol. 108, ed: Springer Berlin Heidelberg, 2012, pp. 77-86.
5. B. Pang and L. Lee, "Opinion mining and sentiment analysis," *Foundations and Trends in Information Retrieval*, vol. 2, pp. 1-135, 2008.

6. M. Hu and B. Liu, "Mining and summarizing customer reviews," in *ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, 2004, pp. 168-17
7. J. Yi and W. Niblack, "Sentiment Mining in WebFountain," in *IEEE International Conference on Data Engineering (ICDE)*, 2005, pp. 1073-1083.
8. P. D. Turney, "Thumbs up or thumbs down?: Semantic orientation applied to unsupervised classification of reviews," in *Association for Computational Linguistics Conference (ACL)*, 2002, pp. 417-424.
9. B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?: sentiment classification using machine learning techniques," in *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2002, pp. 79-86.
10. J. Kamps, M. Marx, R. J. Mokken, and M. De Rijke, "Using wordnet to measure semantic orientations of adjectives," in *International Conference on Language Resources and Evaluation (LREC)* 2004, pp. 1115-1118.
11. A. M. Popescu and O. Etzioni, "Extracting product features and opinions from reviews," in *Conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT/EMNLP)*, 2005, pp. 339-346.
12. S. Morinaga, K. Yamanishi, K. Tateishi, and T. Fukushima, "Mining product reputations on the Web," in *ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, 2002, pp. 341-349.
13. T. Hashimoto and Y. Shirota, "Semantics extraction from social computing: a framework of reputation analysis on buzz marketing sites," in *International Conference on Databases in Networked Information Systems (DNIS)*, 2010, pp. 244-255.
14. S. Baccianella, A. Esuli, and F. Sebastiani, "Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining," in *International Conference on Language Resources and Evaluation (LREC)*, 2010, pp. 2200-2204.

Building Word-Emotion Mapping Dictionary for Online News

Yanghui Rao, Xiaojun Quan, Liu Wenyin, Qing Li, and Mingliang Chen

Department of Computer Science, City University of Hong Kong

Abstract. Sentiment analysis of online documents such as news articles, blogs and microblogs has received increasing attention. We propose an efficient method of automatically building the word-emotion mapping dictionary for social emotion detection. In the dictionary, each word is associated with the distribution on a series of human emotions. In addition, three different pruning strategies are proposed to refine the dictionary. Experiment on the real-world data sets has validated the effectiveness and reliability of the method. Compared with other lexicons, the dictionary generated using our approach is more adaptive for personalized data set, language-independent, fine-grained, and volume-unlimited. The generated dictionary has a wide range of applications, including predicting the emotional distribution of news articles and tracking the change of social emotions on certain events over time.

Keywords: Social emotion detection; emotion dictionary; maximum likelihood estimation

1 Introduction

In the traditional society, when we make a decision, opinions and emotions of others have always been important information for reference. Knowing the answer of "What others think and feel" is usually very necessary for general people, marketers, public relations officials, politicians and managers.

Nowadays, everyone can express their opinions and emotions easily through news portals, blogs and microblogs, and they become both the listeners and speakers. Facing the vast amount of data, tasks of automatically detecting public emotions evoked by online documents is emerging recently [1], such as the SemEval task 14. This task is treated as a classification problem according to the polarity (positive, neutral or negative) or multiple emotion categories such as joy, sadness, anger, fear, disgust and surprise. However, due to the limited information in the news titles, annotating news headlines for emotions is a hard task. It is usually intractable to annotate headlines consistently even for human [2]. As a result, we mainly focus on annotating news bodies for emotions, and building word-emotion mapping dictionaries in this paper.

In previous works, emotions are mostly annotated based on the existing emotional lexicons [1] [3], e.g., Subjectivity Wordlist [4], WordNet-Affect [5] and

SentiWordNet [6]. Emotion classification or opinion mining based on these existing lexicons have their limited utility, because 1) the lexicons are mainly for public use in general domains, some resulting classifications of words can appear incorrect, and need to be adjusted to fit the personalized data set. 2) Most of the lexicons are available only for bits of languages, such as English, and the volume of words annotated is restricted, which limits the applicability of these methods. 3) Some of the lexicons label words on coarse-grained dimensions (positivity, negativity and neutrality), which are insufficient to individuate the whole spectrum of emotional concepts [5].

Unlike the above methods, we focus on building emotional dictionary automatically, in which each item is scored along a number of predefined emotions. Then, the emotion distributions of current news article are estimated accurately based on the emotional dictionary. The main contributions are as follows:

- A method of building the word-emotion mapping dictionary is proposed, which is efficient, precise and automatic, no human resource is needed.
- Three kinds of parameter-free pruning algorithms are presented to refine the dictionary, and to improve the performance.
- Compared with the existing emotional lexicons, the emotional dictionary constructed in this paper is more adaptively for personalized data set, language-independent, fine-grained, and can be updated constantly.

Related works are given in Section 2. The problem definition, the method of building the word-emotion mapping dictionary, pruning algorithms and potential applications of the dictionary are presented in Section 3. The experimental data sets, evaluation metrics, results and discussions are illustrated in Section 4. Finally, we draw conclusions and discuss future work in Section 5.

2 Related Work

Most of the previous works focus on constructing the emotional lexicons for reviews, which is different with ours for news articles. The main features of reviews and news articles are as follows:

For the former data set, people usually explicitly express their opinions and emotions in the reviews, which results in the subjective text; while for the latter data set, news editor normally present the events objectively in the news reports, and their opinions and emotions are transmitted implicitly. In other words, the former data set mainly contains subjective sentences, which express some personal feelings, opinions, views, emotions, or beliefs; while the latter data set mainly contains objective sentences, which present some factual information. Besides, for the former data set, as there exist fraudulent reviews or rumors, the emotional dictionary maybe incorrect or biased; while for the latter data set, the news reports are mainly objective and do not trigger the same problem.

Works of sentiment analysis for reviews rose from the year 2001 or so. Das and Chen [7] utilized classification algorithm to extract market emotions from stock message boards, which was further used for decision on whether to buy or sell a stock. However, the performance heavily depended on certain words. For instance, the sentence “It is not a bear market” means a bull market actually, because negation words such as “no”, “not” are much more important and serve to reverse meaning. Turney [8] applied an unsupervised learning technique to classify the emotional orientation of users’ reviews (such as reviews of movies, travel destinations, automobiles and banks), in which the mutual information differences between each phrase and the words “excellent” and “poor” were calculated firstly. Then, the average emotional orientation of the phrases in the review was used to classify the review as recommended or not recommended.

During this incipient stage of research on sentiment analysis from reviews, some of them focus on using linguistic heuristics or a set of seed words preselected, to classify the emotional orientation of words or phrases [9]. Other works focus on emotional categorization of entire documents, which are based on the construction of discriminate-word dictionaries manually or semi-manually [7]. However, previous experiments shown that the intuition of selecting discriminating words may not always be the best for humans [10]. Besides classifying emotions to positive or negative, predicting the rating scores of reviews has also been done by researchers [11] [12]. As the rating scores are ordinal (e.g., 1-5 stars), the problem is tackled by regression. These previous works of sentiment analysis from reviews are often performed on document, sentence, entity, and feature/aspect level. Emotion classification at both the document and sentence levels is useful, but it cannot find what aspects people liked or disliked. Aspect-based emotional analysis is proposed to tackle such problem, but it is hard to perform on news articles, in which aspects of entity are unknown.

Works of emotion classification for news began from the SemEval tasks in 2007. Chaumartin [1] utilized a linguistic and rule-based approach to tag news headlines for predefined emotions, which includes joy, sadness, anger, fear, disgust and surprise, and for polarity, i.e. positive or negative. The algorithm was based on existing emotional dictionaries, like WordNet-Affect and SentiWordNet. Kolya et al. [3] identified event and emotional expressions at word level from the sentences of TempEval-2010 corpus, in which the emotional expressions are also identified simply based on the sentiment lexicons, e.g., Subjectivity Wordlist, WordNet-Affect and SentiWordNet.

These approaches based on public emotional dictionaries needed extra effort of preprocessing and post-processing on individual words, because some resulting classifications of words can appear incorrect, and need to be adjusted to fit the personalized data set. Katz et al. [2] scored the emotions of each word as the average of the emotions of every news headline, in which that word appears, all non-content words were ignored. However, as the limited words in the news titles, it faced the problem of the small number of words available for the analysis.

In this paper, we mainly focus on annotating news bodies for emotions, and building emotional dictionary automatically. The emotion expressions are fine-grained (such as moving, sympathy, boring, angry and funny), rather than coarse-grained (positive, negative and neutral). The dictionary can be used to classify the emotional distributions of previous unseen news articles.

3 Word-Emotion Mapping Dictionary Construction

In this section, we will firstly define our research problem. Then, we introduce the generation method of the word-emotion mapping dictionary, as well as the pruning algorithms of the generated dictionary. Finally, we discuss the potential applications of the dictionary.

3.1 Problem Definition

The research problem is defined as follows.

Given N training news articles, a word-emotion mapping dictionary is generated. The dictionary is a $W \times E$ matrix, and the $(j,\, k)$ item in this matrix is the score (probability) of emotion e_k conditioned on word w_j.

For each document $d_i(i = 1, 2, ..., N)$, the news content, the publication date (timestamp), and the distribution of ratings of emotions in the predefined list (see Fig. 1 as an example) are available. From these news contents, a vocabulary is obtained as the source of the word-emotion mapping dictionary. The j-th word in the vocabulary is denoted by $w_j(j = 1, 2, ..., W)$, all the emotions is denoted by $e = (e_1, e_2, ..., e_E)$, the normalization form of ratings of d_i over e is denoted by r_i. $r_i = (r_{i1}, r_{i2}, ..., r_{iE})$, and $|r_i| = 1$.

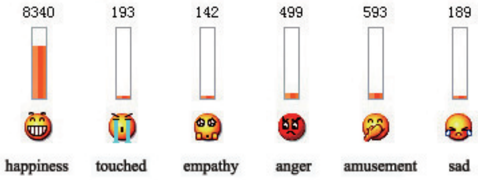


Fig. 1. An example of social emotions and user ratings

3.2 Generation Method

In this section, we introduce the method of generating word-emotion mapping dictionary based on maximum likelihood estimation and the Jensen's inequality.

For each document d_i, the probability of r_i conditioned on d_i can be modeled as:

$$P(r_i|d_i) = \sum_{j=1}^{W} P(w_j|d_i)P(r_i|w_j) \; . \tag{1}$$

Where, the probability of r_i conditioned on w_j is a multinomial distribution, and $P(r_i|w_j) = \prod_{k=1}^{E} P(e_k|w_j)^{r_{ik}}$. Then,

$$P(r_i|d_i) = \sum_{j=1}^{W} P(w_j|d_i) \prod_{k=1}^{E} P(e_k|w_j)^{r_{ik}} \ . \tag{2}$$

In the above, words in document d_i are assumed to be independent.

Let $\sigma_{ij} = P(w_j|d_i)$ and $\theta_{jk} = P(e_k|w_j)$, the log-likelihood over all the N documents can be defined as:

$$\log l = \log(\prod_{i=1}^{N}(\sum_{j=1}^{W} \sigma_{ij} \prod_{k=1}^{E} \theta_{jk}^{r_{ik}})) = \sum_{i=1}^{N} \log(\sum_{j=1}^{W} \sigma_{ij} \prod_{k=1}^{E} \theta_{jk}^{r_{ik}}) \ . \tag{3}$$

According to Jensen's inequality, we reconstruct the log-likelihood as follows:

$$\log l \geq \sum_{i=1}^{N} \sum_{j=1}^{W} \sigma_{ij} \sum_{k=1}^{E} r_{ik} \log \theta_{jk} \ . \tag{4}$$

Since $\sum_{k=1}^{E} \theta_{jk} = 1$, we add a Lagrange multiplier to the log-likelihood equation as follows:

$$\hat{l} = \sum_{i=1}^{N} \sum_{j=1}^{W} \sigma_{ij} \sum_{k=1}^{E} r_{ik} \log \theta_{jk} + \lambda(\sum_{k=1}^{E} \theta_{jk} - 1) \ . \tag{5}$$

Then, we maximize the likelihood by calculating the first-order partial derivative of θ_{jk},

$$\frac{\partial \hat{l}}{\partial \theta_{jk}} = \sum_{i=1}^{N} \frac{\sigma_{ij} r_{ik}}{\theta_{jk}} + \lambda = \frac{\sum_{i=1}^{N} \sigma_{ij} r_{ik}}{\theta_{jk}} + \lambda = 0 \ . \tag{6}$$

Thus,

$$\theta_{jk} = -\frac{\sum_{i=1}^{N} \sigma_{ij} r_{ik}}{\lambda} \ . \tag{7}$$

Since $\sum_{k=1}^{E} \theta_{jk} = 1$, we have

$$\lambda = -\sum_{k=1}^{E} \sum_{i=1}^{N} \sigma_{ij} r_{ik} \ . \tag{8}$$

Then, substitute formula (8) into formula (7) and get

$$\theta_{jk} = \frac{\sum_{i=1}^{N} \sigma_{ij} r_{ik}}{\sum_{k=1}^{E} \sum_{i=1}^{N} \sigma_{ij} r_{ik}} \ . \tag{9}$$

i.e.,

$$P(e_k|w_j) = \frac{\sum_{i=1}^{N} P(w_j|d_i) r_{ik}}{\sum_{k=1}^{E} \sum_{i=1}^{N} P(w_j|d_i) r_{ik}} \ . \tag{10}$$

In the above, $P(e_k|w_j)$ is the probability of emotion e_k conditioned on word w_j from which we can generate the word-emotion mapping dictionary. r_{ik} is the distribution of ratings of document d_i on emotion e_k, $P(w_j|d_i)$ is the probability of word w_j conditioned on document d_i which can be calculated by relative term frequency. The relative term frequency is the number of occurrences of the term w_j in d_i divide by the total number of occurrences of all the terms in d_i.

3.3 Pruning Algorithm

As the size of the training data set increases, the scale of the dictionary extends, making it hard for us to maintain and utilize. Thus, pruning operation is necessary for such lexicons. We will give the definition of *background word* firstly, and then illustrate how it can be used to prune the dictionary.

Definition: Background word is the word that appears in most of the documents in the training data set, it is general for specific domains and topics of the training set, which is quite different with stop words for general domains.

In the context of emotional annotation, the *background words* are general words that contain little emotional information actually and will disturb the effect of utilizing the dictionary. In contrast to other useful emotional tagging words, the probability of a word being to *background words*, which is denoted by $P(B|w)$, is larger than the probability of the word being to emotions, which is denoted by $P(E|w)$. According to the definition, the probability $P(B|w)$ can be represented as follows:

$$P(B|w) = \frac{df_w}{N} \ . \tag{11}$$

In the above, df_w is the document frequency of word w, N is the total number of documents in the training set. The proportion of documents that contains the word w is larger, the probability of w being to *background words* is higher.

As there are multiple emotions tagged for each word according to formula (10), the latter probability $P(E|w)$ has three forms, which are the maximum, average and minimum of all values of $P(e_k|w)$, k is from 1 to E (the total number of types of emotions). Then, the words are pruned from the dictionary if $P(B|w)$ is larger than $P(E|w)$.

When the pruning algorithm above is performed, the word-emotion mapping dictionary is constructed to the end, which can be used to predict the emotions of given news articles as follows:

$$\hat{P}(e|d) = \sum_{w \in W} p(w|d) p(e|w) \ . \tag{12}$$

In the above, $\hat{P}(e|d)$ is the probability of social users having emotions e on document d, $P(w|d)$ is the distribution of new document d on word w, which can be calculated by relative term frequency, $P(e|w)$ is the probability of emotions e conditioned on word w, which can be looked up from the word-emotion mapping dictionary generated with formula (10).

4 Experiments

In this section, experiments are conducted on one Chinese data set and one English data set, so as to test the effect of the word-emotion mapping dictionary on sentiment analysis. The good performances and multilingual data sets reflect the method's effectiveness, reliability, and language-independent of building the dictionary.

4.1 Data Sets

To test the adaptiveness, effectiveness and language-independent of our method of building the word-emotion mapping dictionary, large-scale and multilingual data sets are needed. Two kinds of data sets are employed in the experiment.

Sina. This is a large-scale Chinese data set scrawled from Sina society, which is one of the most popular news sites in China.[1] The attributes include the URL address of the news article, the news headline (title), the publish date (from 29 July, 2005 to 9 Sep, 2011), the news body (content), the user ratings on emotions of touched, empathy, boredom, anger, amusement, sadness, surprise and warmness. The data set contains 32,493 valid news articles with the total number of ratings on the 8 emotions larger than 0. We use x (x = 90%, 80% , 10%) of the data set for training and the remaining (1-x) for testing, to evaluate the scalability and stability of the method.

SemEval. This is an English data set used in the 14th task of the 4th International Workshop on Semantic Evaluations (SemEval-2007).[2] The attributes include the news headline, the score of emotions of anger, disgust, fear, joy, sad and surprise normalizing from 0 to 100. The data set contains 1,246 valid news headlines with the total score of the 6 emotions larger than 0. We use the 1,000 in the test-set (80% of the data set) for training and the 246 in the trial-set (20%) for test.

4.2 Evaluation Metrics

Classifying and predicting the emotions of given news articles are efficient ways to validate the effectiveness of the generated word-emotion mapping dictionary.

The Pearson's correlation coefficient is employed to measure the accuracy of emotion prediction, which indicates the linear dependence between two variables. A value closer to 1 indicates the predicted and the actual emotional distribution fit better, and is reasonable to assert that the trend of ratings on emotions is predicted well by the word-emotion mapping dictionary.

We denote the Pearson's correlation coefficient between the predicted and the actual emotion distributions of the i-th article by pr_i, and the average value of the Pearson's correlation coefficient of all articles by $r\_average$, which is used as the first metric.

[1] http://news.sina.com.cn/society/.

[2] http://nlp.cs.swarthmore.edu/semeval/tasks/.

$$r\_average = \sum_{i=1}^{N} \frac{pr_i}{N} \; . \tag{13}$$

Besides the average value of the Pearson's correlation coefficient, there is another interesting metrics to evaluate the quality of emotion prediction. In practice, when we predicting the multiple emotional distributions, the dominate one with the maximum predicted rating value is attractive.

$$p\_max = \frac{m}{N} \; . \tag{14}$$

In the above, m is the number of articles that the predicted and the actual dominate emotion matched. N is the total number of articles for training or test.

4.3 Results and Analysis

In generating the word-emotion mapping dictionary, the probability of word conditioned on document is calculated by relative term frequency according to formula (10). We denote it by *rtf*. In pruning algorithms, maximum, average and minimum are used to refine the dictionary generated by *rtf* (see section 3.3). We denote these three algorithms by *rtf-max*, *rtf-ave* and *rtf-min*.

Results of Sina For different scales of training data set in Sina, the number of words pruned by *rtf-max*, *rtf-ave* and *rtf-min* are presented in Table 1. The number of the original words in the dictionary ranges from 39,278 to 72,773, within which 45.4% to 31.2% words are pruned using *rtf-min*, the words being pruned are quite less using *rtf-max* and *rtf-ave*.

Table 1. The number of words pruned on Sina

Training documents	*Vocabulary size*	*rtf-max*	*rtf-ave*	*rtf-min*
3,249	39,278	74	302	17,848
6,499	48,555	71	304	18,585
9,748	54,210	67	298	19,185
12,997	58,510	68	295	19,575
16,247	62,105	68	293	20,201
19,496	65,162	67	294	20,858
22,745	67,873	68	296	21,426
25,994	70,447	67	295	22,049
29,244	72,773	67	297	22,672

Fig. 2 depicts the *r_average*, *p_max* of all methods and pruning algorithms on the training and test sets.

For the training set, as the size increases, the quality of emotion prediction decreases at first and then remains stable, from which twofold findings can be observed. The first one lies in that, our dictionary fits the training set well even when the available emotional tagged data is limited. Second, although it is harder to fit the training set when the scale is larger, our dictionary is robust for the stability performance on large training sets. For pruning algorithms, the performances after pruning by *rtf-max*, *rtf-ave* and *rtf-min* are better than others without pruning, among which *rtf-min* performs the best, which shows the significance of our pruning algorithm on refining the dictionary.

For the test set, as the number of test articles increases, the quality of e-motion prediction remains stable mostly, except when the size of test articles is 12,997. This indicates the reliability and stability of the dictionary on predicting emotions of previously unseen articles. For pruning algorithms, the performances by *rtf-max*, *rtf-ave* and *rtf-min* are better than others without pruning, among which *rtf-min* performs the best.

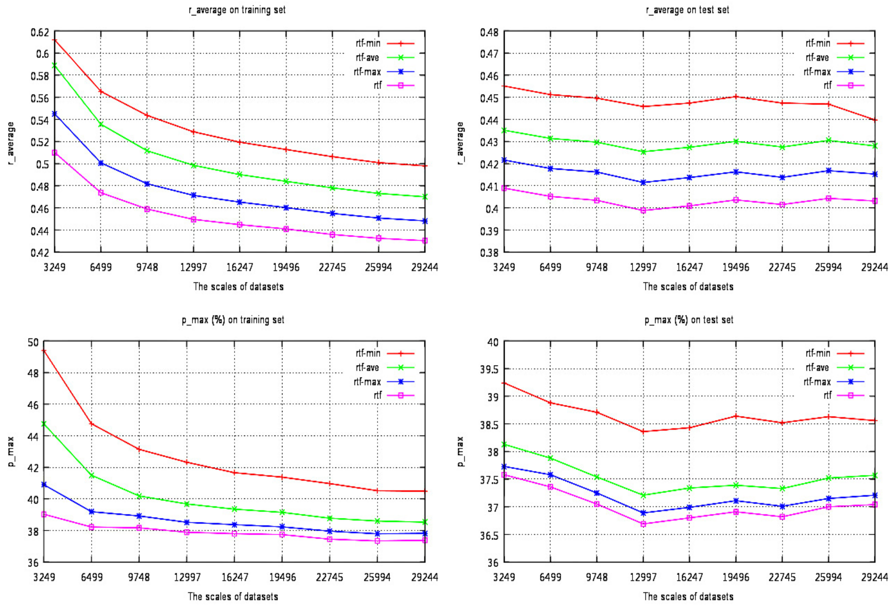


Fig. 2. Performances with different scales of Sina

Although *rtf-min* yields the best results for both training and test data sets, and the improvement over benchmark is remarkable, we also refine the dictionary by deleting the same proportion of words as *rtf-min* randomly, and perform t hypothesis testing on pairwise methods, so as to verify the significant improvement of our pruning algorithm on performances statistically.

The results are depicted in Table 2. For the dictionary after pruning randomly (*prune-random*) and the dictionary without pruning (*rtf*), all of the significance values are much larger than the conventional significance level 0.05, which indicates the dictionary after pruning randomly is no significant different with the dictionary without pruning. In fact, the quality of *prune-random* on the training data set is worse than *rtf* when the size of training documents is 9,748, 12,997 and 19,496, and the quality between them is approximate for other scales of training documents. These findings are similar on the test data set. On the other hand, for the dictionary after pruning by *rtf-min* and *rtf*, or *rtf-min* and *prune-random*, all of the significance values are below the conventional significance level 0.05, which indicates the dictionary after pruning by our method is significant different with others. In our case, we can infer that the dictionary after pruning by *rtf-min* achieves significant performance improvement on both training and test data sets, while pruning randomly does not get such improvement statistically.

Table 2. P-value of the Statistical Significance Test on Sina

Pairwise	*Data set*	*r_average*	*p_max*
prune-random & rtf	Train	0.945303	0.825707
	Test	0.726320	0.886224
rtf-min & rtf	Train	1.18E-04	0.000723
	Test	2.40E-13	9.38E-10
rtf-min & prune-random	Train	1.14E-04	0.000792
	Test	8.63E-14	7.81E-09

Above all, the word-emotion mapping dictionary is effective on emotion classification and prediction. One of the most interesting observations is that when the dictionary is pruned by *rtf-min*, more than 30% words are deleted, while the performances are much better than others.

Results of SemEval Despite that our focus is mainly on annotating emotions for news bodies with long text, it would be very interesting to evaluate the method and pruning algorithms on emotion prediction for news headlines.

The first observation is that when building the word-emotion mapping dictionary based on the short text, as the sparse of the vector, the prune operation maybe unnecessary. For the 1,000 English news headlines used for training here, the vocabulary size is only 2,380 after stemming while retaining the stop words. When the pruning algorithm is applied, the number of pruned words is 0 for *rtf-max* and *rtf-ave*, which means the pruning operation by maximum and average is unnecessary for the data set. The ratio of pruned words is 68.66% for *rtf-min*, which makes the size of the dictionary even smaller, and 7.30% of the training headlines have no word exists in the dictionary, the ratio is 11.38% for test headlines. As a result, pruning by minimize is unsuitable for the SemEval data set, which contains quite limited words.

The second observation is that our method of generating the dictionary works well on fitting the training set for news headlines. The average correlation coefficient of all training articles is 0.86 using the relative term frequency, which shows a strong positive correlation between the predicted and actual emotion distribution. However, the average correlation coefficient of all test articles is 0.36 using the relative term frequency, which means the precision of the dictionary on predicting the emotion distribution of previous unseen documents is relatively low. The reason is that the volume of the word-emotion mapping dictionary is quite small for the limited information of news headlines.

5 Conclusion

Emotion and opinion mining is useful and meaningful from political, economical, commercial, social and psychological perspectives, the word-emotion mapping dictionary constructed in this paper is the first step to meet the needs. Different from previous methods, our method of building the dictionary is adaptive for personalized data set, volume-unlimited, automatically, language-independent, and fine-grained. The main conclusions are as follows:

First of all, the pruning algorithm is effective in refining the dictionary, and improving the performances of emotion prediction. For three forms of removing *background words*, which are maximum, average and minimum, the last one achieves the largest improvement on the performances, and the improvement is statistically significant under hypothesis testing.

Secondly, as the number of training articles increases, the quality of emotion prediction on training data sets decreases firstly and then remains stable. This indicates that our dictionary fits the training data set well even when the available tagged data is limited. Although it is harder to fit the training data set when the scale is larger, our dictionary is robust for the stability performance on large training sets. As the number of test articles increases, the quality of emotion prediction on test data sets remains stable mostly. This indicates the reliable of the word-emotion mapping dictionary on predicting emotions of previously unseen articles.

Last but not least, for annotating emotions of news headlines, it is unnecessary to prune the dictionary, due to the limited vocabulary in the short text. Thus, researches on emotional annotation for both long and short text are our future focuses.

Acknowledgments. The work described in this paper has been supported by the NSFC Overseas, Hong Kong & Macao Scholars Collaborated Researching Fund (61028003)

References

1. Chaumartin, F.R.: Upar7: A knowledge-based system for headline sentiment tagging. In: 4th International Workshop on Semantic Evaluations, pp. 422–425. Association for Computational Linguistics, Prague (2007)

2. Katz, P., Singleton, M., Wicentowski, R.: Swat-mp: The semeval-2007 systems for task 5 and task 14. In: 4th International Workshop on Semantic Evaluations, pp. 308–313. Association for Computational Linguistics, Prague (2007)
3. Kolya, A., Das, D., Ekbal, A., Bandyopadhyay, S.: Identifying Event-Sentiment Association using Lexical Equivalence and Co-reference Approaches. In: Workshop on Relational Models of Semantics Collocated with ACL, pp.19–27. Portland (2011)
4. Banea, C., Mihalcea R., Wiebe J.: A Bootstrapping Method for Building Subjectivity Lexicons for Languages with Scarce Resources. In: 6th International Conference on Language Resources and Evaluation, Marrakech (2008)
5. Strapparava, C., Valitutti, A.: Wordnet-affect: an affective extension of wordnet. In: 4th International Conference on Language Resources and Evaluation, pp. 1083–1086. Lisbon (2004)
6. Baccianella S., Esuli A., Sebastiani F.: SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining. In: 7th Conference on Language Resources and Evaluation, pp. 2200–2204. Valletta (2010)
7. Das, S., Chen, M.: Yahoo! for Amazon: Extracting market sentiment from stock message boards. In: 8th Asia Pacific Finance Association Annual Conference, Shanghai (2001)
8. Turney, P.D.: Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In: 40th annual meeting of the Association for Computational Linguistics, pp. 417–424. Philadelphia (2002)
9. Turney, P.D., Littman, M.L.: Unsupervised learning of semantic orientation from a hundred-billion-word corpus. Technical Report EGB-1094, National Research Council Canada (2002)
10. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up? Sentiment classification using machine learning techniques. In: Empirical Methods in Natural Language Processing, pp. 79–86, Philadelphia (2002)
11. Liu, J., Seneff, S.: Review sentiment scoring via a parse-and-paraphrase paradigm. In: Empirical Methods in Natural Language Processing, ACL, Suntec (2009)
12. Ifrim, G., Weikum, G.: The Bag-of-Opinions Method for Review Rating Prediction from Sparse Text Patterns. In: Coling 2010, Beijing (2010)

Predicting Emotion Labels for Chinese Microblog Texts

Zheng Yuan[1], Matthew Purver[2]

School of Electronic Engineering and Computer Science
Queen Mary University of London
Mile End Road, London E1 4NS
[1]yuanzheng.liliian@hotmail.com
[2]m.purver@qmul.ac.uk

Abstract. We describe an experiment into detecting emotions in texts on the Chinese microblog service Sina Weibo using distant supervision with various author-supplied conventional labels (emoticons and smilies). Existing word segmentation tools proved unreliable; better accuracy was achieved using character-based features. Accuracy varied according to emotion and labelling convention: while smilies are used more often, emoticons are more reliable. Happiness is the most accurately predicted emotion (85.9%). This approach works well and achieves 80% accuracies for "happy" and "fear", even though the performances for the seven emotion classes are quite different.

Keywords: Social Media, Sina Weibo, Emotion Detection, Emoticons, Smilies, Distant Supervision, N-gram lexical features

1 Introduction

Social media has become a very popular communication tool among Internet users. Sina Weibo (hereafter Weibo), is a Chinese microblog website. Most people take it as the Chinese version of Twitter; it is one of the most popular sites in China, in use by well over 30% of Internet users, with a similar market penetration that Twitter has established in the USA (Rapoza, 2011 [1]), and has therefore become a valuable source of people's opinions and sentiments.

Microblog texts (statuses) are very different from general newspaper or web text. Weibo statuses are shorter and more casual; many topics are discussed, with less coherence between texts. Combining this with the huge amount of lexical and syntactic variety (misspelt words, new words, emoticons, unconventional sentence structures) in Weibo data, many existing methods for emotion and sentiment detection which depend on grammar- or lexicon-based information are no longer suitable.

Machine learning via supervised classification, on the other hand, is robust to such variety but usually requires hand-labeled training data. This is difficult and time-consuming with large datasets, and can be unreliable when attempting to infer an author's emotional state from short texts (see e.g. Purver & Battersby, 2012 [2]). Our solution is to use distant supervision: we adapt the approach of (Go et al., 2009 [3]; Purver & Battersby, 2012 [2]) to Weibo data, using emoticons and Weibo's built-in smilies as author-generated emotion labels, allowing us to produce an automatic classifier to classify Weibo statuses into different basic emotion classes. Adapting this approach to Chinese data poses several research problems: finding accurate and reliable labels to use, segmenting Chinese text and extracting sensible lexical features.

Our experiments show that choice of labels has a significant effect, with emoticons generally providing higher accuracy than Weibo's smilies, and that choice of text segmentation method is crucial, with current word segmentation tools providing poor accuracy on microblog text and character-based features proving superior.

2 Background

2.1 Sentiment/Emotion Analysis

Most research in this area focuses on sentiment analysis – classifying text as positive or negative (Pang and Lee, 2008 [4]). However, finer-grained emotion detection is required to provide cues for further human-computer interaction, and is critical for the development of intelligent interfaces. It is hard to reach a consensus on how the basic emotions should be categorised, but here we follow (Chuang and Wu, 2004 [5]) and others in using (Ekman, 1972 [6])'s definition, providing six basic emotions: anger, disgust, fear, happiness, sadness, surprise.

2.2 Distant Supervision

Distant supervision is a semi-supervised learning algorithm that combines supervised classification with a weakly labeled training dataset. (Go et al., 2009 [3]) and (Pak and Paroubek, 2010 [7]), following (Read, 2005 [8]), use emoticons to provide these labels to classify positive/negative sentiment in Twitter messages with above 80% accuracy.

(Yuasa et al., 2006 [9]) showed that emoticons have an important role in emphasizing the emotions conveyed in a sentence; they can therefore give us direct access to the authors' own emotions. (Purver and Battersby, 2012 [2]) thus used a broader set of emoticons to extend the distant supervision approach to six-way emotion classification in English, and we apply a similar approach. However, in addition to the widely used, domain-independent emoticons, other markers have emerged for particular interfaces or domains. Sina Weibo provides a built-in set of smilies that can work as special emoticons that help us better understand authors' emotions.

2.3 Chinese Text Processing

In Chinese text, sentences are represented as strings of Chinese characters without explicit word delimiters as used in English (e.g. white space). Therefore, it is important to determine word boundaries before running any word-based linguistic processing on Chinese. There is a large body of research into Chinese word segmentation (Fan and Tsai, 1988 [10]; Sproat and Shih, 1990 [11]; Gan et al, 1996 [12]; Guo, 1997

[13]; Jin and Chen, 1998 [14]; Wu, 2003 [15]). Among them, the basic technique for identifying distinct words is based on the lexicon-based identification scheme (Chen and Liu, 1992). This approach performs word segmentation process using matching algorithms: matching input character strings with a known lexicon. However, since the real-world lexicon is open-ended, new words are coming out every day – and this is especially true with social media. A lexicon is therefore difficult to construct or maintain accurately for such a domain.

3 Data

3.1 Corpus Collection

Our training data consisted of Weibo statuses with emoticons and smilies. Since Weibo has a public API, training data can be obtained through automated means. We wrote a script which requested the statuses public_timeline API[1] every two minutes and inserted the collected data into a MySQL database. We collected a corpus of Weibo data, filtering out messages not containing emotion labels (see below and Table 2 for details).

3.2 Emotion Labels

We used two kinds of emotion labels (emoticons and smilies) as our noisy labels. The emoticons and smilies are noisy themselves: ambiguous or vague. Not all the emoticons and smilies have close relationships with the emotion classes. And some emoticons and smilies may be used in different situations, as different people have different understandings. Emoticons here are Eastern-style emoticons, very different from Western-style ones (see e.g. Kayan et al., 2006 [16]). Smilies are Sina Weibo's built-in smilies. Initial investigation found that not all emoticons and smilies can be classified into Ekman's six emotion classes; and for some lesser used labels, authors have widely different understandings. We identified the most widely used and well-known emoticons/smilies to use as labels – see Table 1.

Table 1. Conventional markers used for emotion classes

Emotion	Emoticons	Smilies
surprise	OMG; (0.o); (O_o); (@_@); (O_O); (O?O)	😲[吃惊 chi-jing “surprise”]
disgust	N/A	🤮[吐 tu “sick”]

happy	(^_^); (*^__^*);(^o^);
	(^.^);O(∩_∩)O;
		😁[嘻嘻 xi-xi “heehee”];
		😆[哈哈 ha-ha “haha”];
		👏[鼓掌 gu-zhang “applaud”];

[1] http://open.weibo.com/wiki/2/statuses/public_timeline

		[太开心 da-kai-xin “so happy”]

angry	╭(╯^╰)╮ ; (╰_╯)	[怒 nu “anger”];
		[怒骂 nu-ma “curse”];
		[哼 heng “humph”];
		[鄙视 bi-shi “disdain”]

fear	Just use the keyword 害怕 hai-pai “fear”	

sad	(T_T); (T.T); (T^T);
	(ㅠ.ㅠ);
	(╯﹏╰); (︶︿︶);
		[泪 lei “tear”];
		[失望 shi-wang “disappointed”];
		[悲伤 bei-shang “sad”]

3.3 Text Processing

We used a Chinese language selection filter to filter out all other language characters or words, removed URLs, Weibo usernames (starting with @), digits, and any other notations, e.g., *, ￥, only leaving Chinese characters. We then removed the emoticons and smilies from the texts, replacing them with positive/negative labels for the relevant emotion classes for training and testing purposes. We then extracted different kinds of lexical features: segmented Chinese words, Chinese characters, and higher order n-grams.

For word-based features, we need to segment the sentences. There are lots of Chinese word segmentation tools; however, many are unsuitable for online social media text; we chose pymmseg[2], smallseg[3] and the Stanford Chinese Word Segmenter[4], which all appeared to give reasonable results. Pymmseg uses the MMSEG algorithm (Tsai, 2000 [17]). Smallseg is an open sourced Chinese segmentation tool based on DFA. The Stanford Segmenter is CRF-based (Tseng et al, 2005 [18]).

3.4 Corpus Analysis

Our database contains 229,062 Weibo statuses with emotion labels; Table 2 shows statistics. The number of Weibo statuses varied with the popularity of the labels themselves: “happy” and “sad” labels are much more frequent than others; very similar results are observed in English Twitter statuses (see e.g. [2]), suggesting that these frequencies are relatively stable across very different languages.

Table 2. Number of statuses per emotion class

Emotion	Mixed	Emoticons	Smilies

[2] http://code.google.com/p/pymmseg-cpp/

[3] http://code.google.com/p/smallseg/

[4] http://nlp.stanford.edu/software/segmenter.shtml

surprise	347	63	284
disgust	142	N/A	142
happy	5685	712	4973
angry	2318	9	2305
fear	480	Key words: 480	
sad	5422	1064	4358

Overall frequencies show that users of Weibo are more likely to use the built-in smilies rather than emoticons. One possible reason is that smilies can be inserted with a single mouse click, whereas emoticons must be typed using several keystrokes – Eastern-style emoticons are usually made of five or more characters.

4 Experiments and Discussions

Classification was using support vector machines (SVMs) (Vapnik, 1995 [19]) throughout, with the help of the LibSVM tools (Chang and Lin, 2001 [20]). The performance was evaluated using 10-fold cross validation. Our datasets were balanced: a dataset of size N contained N/2 positive instances (statuses containing labels for this emotion class) and N/2 negative ones (statuses containing labels from other classes). For the N/2 negative instances, we randomly selected instances from other emotion classes for larger datasets (N>1200), but ensured an even weighting across negative classes for smaller sets to prevent bias towards one negative class. Because of the different frequency of different emotion labels, we mainly focused on "happy", "angry" and "sad", and present tentative results for the other emotion classes.

4.1 Segmented Words-VS-Characters

In the first experiment, we investigated the effect of different segmentation tools and compared word-based vs character-based features.

After testing on "angry", "happy" and "sad", we found that pymmseg outperformed the other tools; we therefore used pymmseg for later experiments. However, as we increased the dataset size, we found that character-based features had even better performance than word features (using pymmseg) for all three classes. Our results suggest that we could just use Chinese characters, rather than doing any word segmentation - see Figure 1.

Examination of the segmented data showed that the segmentation tools didn't work well with our social media data and made lots of mistakes. In addition, all segmentation tools produced many segmented words which were actually just one character. The use of character-based features was therefore preferred.

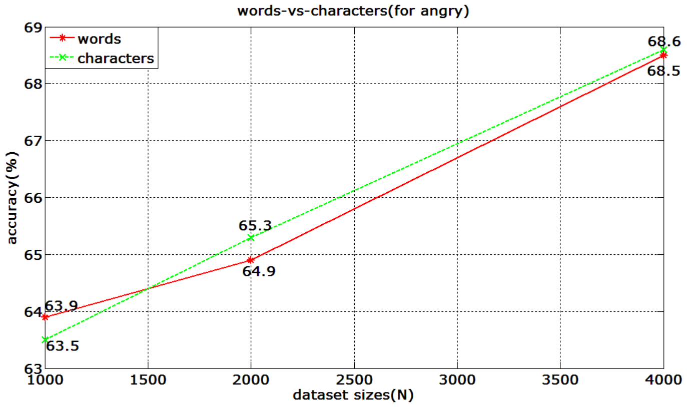


Fig. 1. Words-vs-Characters (for "angry")

4.2 Increasing Accuracy

In the second experiment, we tried to improve the overall performance.

Whether higher-order n-grams are useful features appears to be a matter of some debate. (Pang et al., 2002 [21]) report that unigrams outperform bigrams when classifying movie reviews by sentiment polarity, but (Dave et al., 2003 [22]) find that bigrams and trigrams can give better product-review polarity classification.

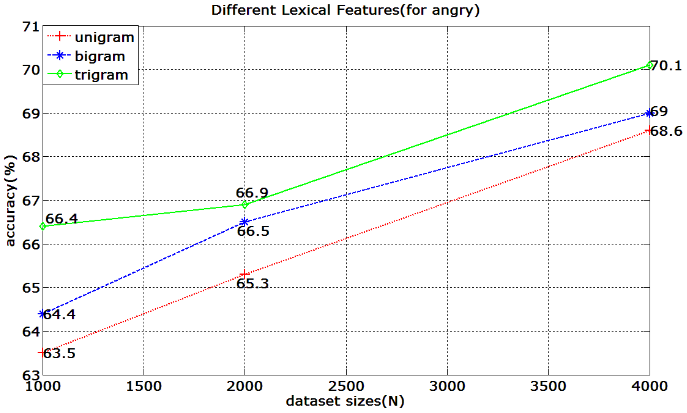


Fig. 2. Performance of n-grams (for "angry")

Results showed that higher-order n-grams are useful features for our wide-topic social media Weibo data. Bigrams and trigrams outperform unigrams for all these three emotion classes (see Figure 2). In our experiments with bigram and trigram features, we also included the lower-order n-grams (unigrams, bigrams), as there are lots of Chinese words with only one character. Our experiments also showed that increasing our dataset sizes increased accuracy; as our dataset sizes increase over time, we therefore expect improvements in accuracy (Figs 1 and 2).

Table 3. Best performance for three emotion classes

Emotion	Dataset Size N	Accuracy
angry	4000	70.1%
happy	8000	78.2%
sad	8000	69.6%

4.3 Smilies-vs-Emoticons

Our last experiment compared the two different kinds of labels: emoticons and smilies.

Table 4. Results of emoticons-vs-smilies (N=1200)

Emotion	Mixed	Emoticons	Smilies
happy	73.8%	85.9%	74.6%
sad	62.8%	67.5%	66.0%

Results showed that the emoticon labels were easier to classify than smilies. By looking at the data, we found that people use emoticons in a more systematic or consistent way. They use emoticons to tell others what their real emotions are ("happy", "sad" etc.), but on the other hand, they use smilies for a much bigger range of things, such as jokes, sarcasm, etc. Some people use smilies just to make their Weibo statuses more interesting and lively, apparently without any subjective feelings.

5 Conclusion

We used SVMs for automatic emotion detection for Chinese microblog texts. Our results show that using emoticons and smilies as noisy labels is an effective way to perform distant supervision for Chinese. Emoticons seem to be more reliable for emotion detection than smilies. It was also found that, when dealing with social media data, many Chinese word segmentation tools do not work well. Instead, we can use characters as lexical features and performance improves with higher-order n-grams. Increasing the dataset size also improves performance, and our future work will examine larger sets.

References

1. Kenneth Rapoza: China's Weibos vs US's Twitter: And the Winner Is? http://www.forbes.com/sites/kenrapoza/2011/05/17/chinas-weibos-vs-uss-twitter-and-the-winner-is/ (2011)
2. Matthew Purver and Stuart Batersby: Experimenting with Distant Supervision for Emotion Classification. In: 13th Conference of the European Chapter of the Association for Computational Linguistics. (2012)
3. Alec Go, Richa Bhayani, and Lei Huang: Twitter sentiment classification using distant supervision. Master's thesis, Stanford University. (2009)

4. Bo Pang and Lillian Lee: Opinion mining and sentiment analysis. Foundations and Trends in In- formation Retrieval, 2(1–2):1–135. (2008)
5. Ze-Jing Chuang and Chung-Hsien Wu: Multi- modal emotion recognition from speech and text. In: Computational Linguistics and Chinese Language, 9(2):45–62. (2004)
6. Paul Ekman: Universals and cultural differ- ences in facial expressions of emotion. In J. Cole, editor, Nebraska Symposium on Motivation 1971, volume 19. University of Nebraska Press. (1972)
7. Alexander Pak and Patrick Paroubek: Twitter as a corpus for sentiment analysis and opinion mining. In: 7th Conference on International Language Resources and Evaluation. (2010)
8. Jonathon Read: Using emoticons to reduce dependency in machine learning techniques for sentiment classification. In: 43rd Meeting of the Association for Computational Linguistics. (2005)
9. Masahide Yuasa, Keiichi Saito and Naoki Mukawa: Emoticons convey emotions without cognition of faces: an fMRI study. CHI EA '06. ISBN: 1-59593-298-4, doi: 10.1145/1125451.1125737 (2006)
10. Fan, C. K., & Tsai, W. H.: Automatic word identification in Chinese sentences by the relaxation technique. Computer Processing of Chinese & Oriental Languages, 4, 33-56. (1988)
11. Richard Sproat and Chilin Shih: A Statistical Method for Finding Word Boundaries in Chinese Text. Computer Processing of Chinese and Oriental Languages, 4, 336-351, (1990)
12. Kok-Wee Gan, , Martha Palmer, and Kim-Teng Lua: A statistically emergent approach for language processing: Application to modeling context effects in ambiguous Chinese word boundary perception. Computational Linguistics, 22(4):531–53. (1996)
13. Jin Guo: Critical tokenization and its properties. Computational Linguistics, 23(4):569–596. (1997)
14. Wangying Jin, and Lei Chen: Identifying unknown words in Chinese corpora. In: First Workshop on Chinese Language, University of Pennsylvania, Philadelphia. (1998)
15. Andi Wu: Customizable segmentation of morphologically derived Words in Chinese. In: Computational Linguistics and Chinese Language. 8(2). (2003)
16. Shipra Kayan, Susan R. Fussell and Leslie D. Setlock: Cultural differences in the use of instant messaging in Asia and North America. In: 20th anniversary conference on Computer supported cooperative work, Banff, Alberta, Canada. (2006)
17. Chih-Hao Tsai: MMSEG: A Word Identification System for Mandarin Chinese Text Based on Two Variants of the Maximum Matching Algorithm. http://technology.chtsai.org/mmseg/ (2000)
18. Huihsin Tseng, Pichuan Chang, Galen Andrew, Daniel Jurafsky and Christopher Manning: A Conditional Random Field Word Segmenter. In: Fourth SIGHAN Workshop on Chinese Language Processing. (2005)
19. Vladimir N. Vapnik: The Nature of Statistical Learning Theory. (1995)
20. Chih-Chung Chang and Chih-Jen Lin: LIBSVM: a library for Support Vector Machines. http://www.csie.ntu.edu.tw/~cjlin/libsvm/ (2001)
21. Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan: Thumbs up? Sentiment classification using machine learning techniques. In: Conference on Empirical Methods in Natural Language Processing, pages 79–86. (2002)
22. Kushal Dave, Steve Lawrence, and David M. Pennock: the peanut gallery: Opinion extraction and semantic classification of product reviews. In: WWW, pages 519–528. (2003)

Feature Weighting Strategies in Sentiment Analysis

Olena Kummer and Jacques Savoy

Rue Emile-Argand 11, CH-2000 Neuchâtel
{olena.zubaryeva,jacques.savoy}@unine.ch
http://www2.unine.ch/iiun

Abstract. In this paper we propose an adaptation of the Kullback-Leibler divergence score for the task of sentiment and opinion classification on a sentence level. We propose to use the obtained score with the SVM model using different thresholds for pruning the feature set. We argue that the pruning of the feature set for the task of sentiment analysis (SA) may be detrimental to classifiers performance on short text. As an alternative approach, we consider a simple additive scheme that takes into account all of the features. Accuracy rates over 10 fold cross-validation indicate that the latter approach outperforms the SVM classification scheme.

Keywords: Sentiment Analysis, Opinion Detection, Kullback-Leibler divergence, Natural Language Processing, Machine Learning

1 Introduction

In this paper we consider sentiment and opinion classification on a sentence level. Sentiment analysis of user reviews, and short text in general could be of interest for many practical reasons. It represents a rich resource for marketing research, social analysts, and all interested in following opinions of the mass. Opinion mining can also be useful in a variety of other applications and platforms, such as recommendation systems, product ad placement strategies, question answering, and information summarization.

The suggested approach is based on a supervised learning scheme that uses feature selection techniques and weighting strategies to classify sentences into two categories (opinionated vs. factual or positive vs. negative). Our main objective is to propose a new weighting technique and classification scheme able to achieve comparable performance to popular state-of-the-art approaches, and to provide a decision that can be understood by the final user (instead of justifying the decision by considering the distance difference between selected examples).

The rest of the article is organized as follows. First, we present the review of the related literature in Section 2. Next, we present the adaptation of the Kullback-Leibler divergence score for opinion/sentiment classification in Section 3. Section 4 provides a description of the experimental setup and corpora used. Sections 5 and 6 present experiments and analysis of the proposed weighting measure with the SVM model, and additive classification scheme respectively. Finally, we give conclusions in Section 7.

2 Related Literature

Often as a first step in machine learning algorithms, like SVM, naïve Bayes, k-Nearest Neighbors, one uses feature weighting and/or selection based on the computed weights. The selection of features allows decrease of the dimensionality of the feature space and thus the computational cost. It can also reduce the overfitting of the learning scheme to the training data. Several studies expose the feature selection question. Forman [1] reports an extensive evaluation of various schemes in text classification tasks. Dave *et al.* [2] give an evaluation of linguistic and statistical measures, as well as weighting schemes to improve feature selection.

Kennedy *et al.* [3] use General Inquirer [4] to classify reviews based on the number of positive and negative terms that they contain. General Inquirer assigns a label to each sense of the word out of the following set: *positive*, *negative*, *overstatement*, *understatement*, or *negation*. Negations reverse the term polarity while overstatement and understatements intensify or diminish the strength of the semantic orientation.

In the study carried out by Su *et al.* [5] on MPQA (Multi-Perspective Question Answering) and movie reviews corpora it is shown that publicly available sentiment lexicons can achieve the performance on par with the supervised techniques. They discuss opinion and subjectivity definitions across different lexicons and claim that it is possible to avoid any annotation and training corpora for sentiment classification. Overall, it has to be noted that opinion words identified with the use of the corpus-based approaches may not necessarily carry the opinion itself in all situations. For example, *He is looking for a good camera on the market.* Here, the word *good* does not indicate that the sentence is opinionated or expresses a positive sentiment.

Pang *et al.* [6] propose to first separate subjective sentence from the rest of the text. They assume that two consecutive sentences would have similar subjectivity label, as the author is inclined not to change sentence subjectivity too often. Thus, labeling all sentences as objective and subjective they reformulate the task of finding the minimum s-t cut in a graph. They carried out experiments on the movie reviews and movie plot summaries mined from the Internet Movie DataBase (IMDB), achieving an accuracy of around 85%.

A variation of the SVM method was adopted by Mullen *et al.* [7] who use WordNet syntactic relations together with topic relevance to calculate the subjectivity scores for words in text. They report an accuracy of 86% on the Pang *et al.* [8] movie review dataset. An improvement of one of the IR metrics is proposed in [9]. The so-called "Delta TFIDF" metric is used as a weighting scheme for features. This metric takes into account how the words are distributed in the positive vs. negative training corpora. As a classifier, they use SVM on the movie review corpus.

Paltoglou *et al.* [10] explore IR weighting measures on publicly available movie review datasets. They have good performance with BM25 and smoothing, showing that it is important to use term weighting functions that scale sublineary in relation to a number of times a term occurs in the document. They underline that the document frequency smoothing is a significant factor.

3 KL Score

In our experiments we adopted a feature selection measure described in [11] that is based on the Kullback-Leibler divergence (KL-divergence) measure. In this paper, the author seeks to find a measure that would lower the score of the features that have different distribution in the individual training documents of a given class from the distribution in the whole corpus. Thus, the scoring function would allow to select features that are representative of all documents in the class leading to more homogeneous classes. The scoring measure based on KL-divergence introduced in [11] yields an improvement over MI with naïve Bayes on Reuters dataset, frequently used as a text classification benchmark.

Schneider [11] shows how we can use the KL-divergence of a feature f_t over a set of training documents $S = d_1, ..., d_{|S|}$ and classes c_j, $j = 1, ..., |C|$ is given in the following way:

$$KL_t(f) = \tilde{K_t}(S) - \tilde{KL_t}(S) \tag{1}$$

where $\tilde{K_t}(S)$ is the average divergence of the distribution of f_t in the individual training documents from all training documents. The difference $KL_t(f)$ in the Equation 1 is bigger if the distribution of a feature f_t is similar in the documents of the same class and dissimilar in documents of different classes.

$\tilde{K_t}(S)$ is defined in the following way:

$$\tilde{K_t}(S) = -p(f_t) \log q(f_t) \tag{2}$$

where $p(f_t)$ is the probability of occurrence of feature f_t (in the training set). This probability could be estimated as the number of occurrences of f_t in all training documents, divided by the total number of features. Let N_{jt} be the number of documents in c_j that contain f_t, and $N_t = \sum_{j=1}^{|C|} N_{jt}/|S|$. Then $q(f_t|c_j) = \sum_{j=1}^{|C|} N_{jt}/|c_j|$ and $q(f_t) = N_t/|S|$. The second term from 1 is defined as follows:

$$\tilde{KL_t}(S) = -\sum_{j=1}^{|C|} p(c_j) p(f_t|c_j) \log q(f_t|c_j) \tag{3}$$

where $p(c_j)$ is the prior probability of category c_j, and $p(f_t|c_j)$ is the probability that the feature f_t appears in a document belonging to the category c_j. Using the maximum likelihood estimation with a Laplacean smoothing, Schneider [11] obtains:

$$p(f_t|c_j) = \frac{1 + \sum_{d_i \in c_j} n(f_t, d_i)}{|V| + \sum_{t=1}^{|V|} \sum_{d_i \in c_j} n(f_t, d_i)} \tag{4}$$

where $|V|$ is the training vocabulary size or the number of features indexed, $n(f_t, d_i)$ is the number of occurrences of f_t in d_i. It is important to note that the afore mentioned average diversion calculations are really approximations based on two assumptions: the number of occurrences of f_t is the same in all documents containing f_t, and all documents in the class c_j have the same length. These two assumptions may turn detrimental for long extract text classification as noted by the author himself [11], but turn out quite effective for a sentence classification setup where a phrase mostly consists of features that occur once, with usually low variations in sentence length. It is important to note that the computation of $p(f_t|c_j)$ should be done on a feature set with removed outliers, since they occur in all or almost all sentences in the corpora.

In sentence-based classification the pruning of the feature set can turn out quite detrimental to the classification accuracy. This is true if the size of the training set is not big enough in order to be sure that some important for classification features are not discarded. Thus, we propose to modify the KL-divergence measure for sentiment and opinion classification. In [11] it calculates the difference between the average divergence of the distribution of f_t in individual training documents from the global distribution, all this averaged over all training documents in all classes. For the sentiment/opinion classification task it is interesting to calculate the difference between the average divergence in one class from the distribution over all classes. Therefore, we can obtain the average divergence of the distribution of f_t for each of the classification categories ($j \in POS, NEG$):

$$\tilde{KL_t}^{j}(S) = N_{jt} \cdot \tilde{p}_d(f_t|c_j) log \frac{\tilde{p_d}(f_t|c_j)}{p(f_t|c_j)} \tag{5}$$

Substituting $\tilde{KL_t}^{POS}(S)$ and $\tilde{KL_t}^{NEG}(S)$ in Equation 1 for each category we obtain measures that evaluate how different is the distribution of feature f_t in one category from the whole training set.

$$KL_t^{POS}(f) = \tilde{K_t}^{POS}(S) - \tilde{KL_t}^{POS}(S) \tag{6}$$

$$KL_t^{NEG}(f) = \tilde{K_t}^{NEG}(S) - \tilde{KL_t}^{NEG}(S) \tag{7}$$

This way, we obtain two sums $\sum KL_t^{POS}(f)$ and $\sum KL_t^{NEG}(f)$ over the features present in the sentence. The final difference of the two sums (denoted further as KL score) can serve as a prediction score of to which category the sentence is most similar.

4 Experimental Setup and Dataset Description

We use the setup with unigram indexing, short stop word elimination (several prepositions and verb forms: *a, the, it, is, of*) and the use of the Porter stemmer [12]. All reported experiments use 10 fold cross-validation setup.

In our study we use three publicly available datasets that we chose based on their popularity as benchmark datasets in SA research. The first one is Sentence Polarity dataset v1.0[1] [13]. It contains 5331 positive and 5331 negative snippets of movie reviews, each review is one sentence long. The Subjectivity dataset contains 5000 subjective and 5000 objective sentences [6].

As a third dataset, that contains newspaper articles, we use the MPQA dataset[2] [14]. The problem with the MPQA dataset is that the annotation unit is at the phrase level, which could be a word, part of a sentence, a clause, a sentence itself, or a long phrase. In order to obtain a dataset with a sentence as a classification unit, we used the approach proposed by Wilson *et al.* [14]. They define the sentence-level opinion classification in terms of the phrase-level annotations. A sentence is considered opinionated if:

1. It contains a "GATE_direct-subjective" annotation with the attribute intensity not in ['low', 'neutral'] and not with the attribute 'insubstantial';
2. The sentence contains a "GATE_expressive-subjectivity" annotation with attribute intensity not in ['low'].

Here is the information on corpus statistics as reported in [14]: there are 15,991 subjective expressions from 425 documents, containing 8,984 sentences. After parsing, we obtained 6,123 opinionated and 4,989 factual sentences.

5 KL Score and SVM

We were interested in evaluating the features selected by our method with the use of the SVM classifier. As pointed out in [15], SVM is able to learn a model independent of the dimension of the space with few irrelevant features present. The experiments on text categorization task show that even the features, that are ranked low according to their IG, are still relevant and contain the information needed for successful classification. Another particularity of the text classification tasks in the context of the SVM method is the sparsity of the input vector, especially when the input instance is a sentence, and not a document.

Joachims [15] observed that the text classification problems are usually linearly separable. Thus, a lot of the research dealing with text classification uses linear kernels [1]. In our experiments we used SVM^{light} implementation with the linear kernel with the soft-margin constant *cost* = 2.0 [16]. We chose *cost* value based on the experimental results. Generally, the low *cost* value (by default 0.01) indicates a bigger error tolerance during training. With the growth of the *cost* value the SVM model assigns larger penalty for margin errors.

We also experimented with other types of kernels, namely with the radial basis function kernel. From our experiments, learning of the SVM model with this kernel takes substantially longer time and gives approximately the same level of the performance as the linear kernel.

[1] http://www.cs.cornell.edu/people/pabo/movie-review-data

[2] http://www.cs.pitt.edu/mpqa/

Dataset	**% of features**		
	60%	*80%*	*100%*
Polarity	65.93	65.93	65.88
Subjectivity	84.72	84.72	84.69
MPQA	68.42	68.42	68.46

Table 1. Accuracy of SVM^{light} with the linear kernel ($\gamma = 2.0$) and different percentage of features.

We prune the ranked features by the score, accounting for at least 60% of the feature set. This is due to the fact that further pruning of features leads to drastic degradation in accuracy. Further elimination of features from the training model leads to the situation when some testing sentences are represented with one or two features only. The pruning of the feature set up to 60% and 80% of top ranked features did not ameliorate the accuracy of the KL score and the SVM model. In the next section, we discuss possible reasons for degradation in accuracy when pruning the feature set for the task of SA classification on a sentence level, and propose a simple additive classification scheme.

6 Additive Classification Model Based on KL Score

In text classification, after calculating the scores between every feature and every category the next steps are to sort the features by score, choose the best k features and use them later to train the classifier. For the task of sentiment classification on a sentence-based level the pruning of the feature set may lead to the elimination of infrequent features (several occurrences) and may cause the loss of important information needed for classification of the new instances. Here are some differences in the aspects of use of the feature selection measures in text classification and opinion/sentiment analysis contexts. First, the aim in topic text classification is to look for the set of topic-specific features that describe the classification category. In sentiment classification, though, the markers of the opinion could be carried by both topic-specific and context words that may also have small differences in distributions across categories due to the short text length. If we look at the opinion review domain, the topic-specific features would be *movie*, *film*, *flick* and context words would be (*long*, *short*, *horror*, *satisfy*, *give up*).

Second, the usual text classification methods are designed for documents consisting of at least several hundreds of words, assuming that the features that could aid in classification repeat across the text several times. The format of a sentence does not let us make the same assumption. The opinion or sentiment polarity can be expressed with the help of one word/feature. There is substantial evidence from several studies that the presence/absence of a feature is a better indicator than the tf scores [17].

Thus, for effective classification, the model should identify features that are strong indicators of opinion/sentiment, take into account the relations between the features in each category, and be able to adjust scores of the features that were not frequent enough in order to expand the set of features that are strong indicators of the sentiment.

As a classification model we use a simple additive score of the features in the sentence computed for each category. Our aim is to determine the behavior of the KL score for the task of sentence sentiment and opinion classification in terms of its goodness and priority in feature weighting based on feature distribution across classification categories.

Dataset	**Prec.**	**Recall**	**F1**	**Acc.**
Polarity	67.26%	72.01%	69.55%	68.48%
Subjectivity	91.17%	90.64%	90.90%	90.93%
MPQA	75.53%	61.39%	67.69%	65.07%

Table 2. Precision, recall, F1-measure, and accuracy of all metrics over the three corpora: Movie Review, Subjectivity and MPQA datasets.

From the results presented in the Table 2, we can see that a simple classification scheme based on computing the sum of the feature scores according to the classification category outperforms the SVM model on the sentence datasets. As we deal with a small number of features, it is advantageous to use all of them when taking a classification decision. Comparing with the results in Table 1, we have achieved an improvement in accuracy for the Polarity and Subjectivity datasets. Nevertheless, the SVM model gives better results for the MPQA corpus. This may be due to the stylistics and opinion annotation and expression differences in movie and newspaper domains. The former is usually much more expressive, containing more sentiment-related words, than the latter.

7 Conclusions

In this article we suggest a new adaptation of the Kullback-Leibler divergence score as a weighting measure for sentiment and opinion classification. The proposed score, named KL score, we use for feature weighting with the SVM model. The experiments showed that the pruning of the feature set does not improve the SVM performance. Taking into account the differences in topical and sentiment classification of short text, we proposed a simple classification scheme based on calculation of sum of the features present in the sentence according to each classification category. Surprisingly, this scheme yields better results than SVM.

Based on the three well-known test-collections in the domain (Sentence Polarity, Subjectivity and MPQA datasets), we suggested a new way of computing feature weights, that could be later used with SVM or other supervised classification schemes that use feature weight computation. The proposed score and classification model were successfully applied in two different contexts (sentiment and opinion) and two domains (movie review and news articles).

References

1. Forman, G.: An extensive empirical study of feature selection metrics for text classification. The Journal of Machine Learning Research, Special Issue on Variable and Feature Selection, vol. 3, pp. 1289–1305. (2003)
2. Dave, K., Lawrence, S., Pennock, D. M.: Mining the peanut gallery: opinion extraction and semantic classification of product reviews. In Proceedings of the WWW Conference, pp. 519–528. (2003)
3. Kennedy, A., Inkpen, D.: Sentiment classification of movie reviews using contextual valence shifters. In Journal of Computational Intelligence, vol. 22(2), pp. 110–125. (2006)
4. Stone, P.J.: The General Inquirer: a computer approach to content analysis. The MIT Press. (1966)
5. Su, F., Markert, K.: From words to senses: a case study of subjectivity recognition. In Proceedings of the 22nd International Conference on Computational Linguistics, pp. 825–832. (2008)
6. Pang, B., Lee, L.: A sentimental education: sentiment analysis using subjectivity summarization based on Minimum Cuts. In Proceedings of the 42nd Annual Meeting of the ACL. (2004)
7. Mullen, T., Collier, N.: Sentiment analysis using Support Vector Machines with diverse information sources. In Proceedings of the EMNLP Conference, pp. 412–418. (2004)
8. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up?: sentiment classifcation using machine learning techniques. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 79–86. (2002)
9. Martineau, J., Finin, T.: Delta TFIDF: an improved feature space for sentiment analysis. In Proceedings of the AAAI Conference on Weblogs and Social Media. (2009)
10. Paltoglou, G., Thelwall, M.: A study of information retrieval weighting schemes for sentiment analysis. In Proceedings of the 48th Annual Meeting of the ACL, pp. 1386–1395. (2010)
11. Schneider, K.M.: A new feature selection score for multinomial naïve Bayes text classification based on KL-divergence. Proceedings of the 42nd Annual Meeting of the ACL. (2004)
12. Porter, M. F.: Readings in information retrieval. Morgan Kaufmann Publishers Inc. (1997)
13. Pang, B., Lee, L.: Seeing stars: exploiting class relationships for sentiment categorization with respect to rating scales. In Proceedings of the 43rd Annual Meeting of ACL, pp. 115–124. (2005)
14. Wilson, T., Wiebe, J., Hoffmann, P.: Recognizing contextual polarity in phrase-level sentiment analysis. In Proceedings of the HLT and EMNLP, pp. 354–362. (2005)
15. Joachims, T.: Text categorization with Support Vector Machines: learning with many relevant features. In Proceedings of the European Conference on Machine Learning, pp. 137–142. (1998)
16. Joachims, T.: Making large-scale (SVM) learning practical. Advances in Kernel Methods - Support Vector Learning, pp. 169–184. MIT Press, Cambridge, MA. (1999)
17. Pang, B., Lee, L.: Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval, vol. 2(1-2). (2008)

Sentimentor: Sentiment Analysis of Twitter Data

James Spencer and Gulden Uchyigit

School of Computing, Engineering and Mathematics
University of Brighton,
Brighton, BN2 4GJ
{j.spencer1,g.uchyigit}@brighton.ac.uk

Abstract. In this paper we present Sentimentor, a tool for sentiment analysis of Twitter data. Sentimentor utilises the naive Bayes Classifier to classify Tweets into positive, negative or objective sets. We present experimental evaluation of our dataset and classification results, our findings are not contridictory with existing work.

Keywords: sentiment analysis, opinion mining, classification, machine learning

1 Introduction

Social networks have revolutionised the way in which people communicate. Information available from social networks is beneficial for analysis of user opinion, for example measuring the feedback on a recently released product, looking at the response to policy change or the enjoyment of an ongoing event. Manually sifting through this data is tedious and potentially expensive.

Sentiment analysis is a relatively new area, which deals with extracting user opinion automatically. An example of a positive sentiment is, *"natural language processing is fun"* alternatively, a negative sentiment is *"it's a horrible day, i am not going outside"*. Objective texts are deemed not to be expressing any sentiment, such as news headlines, for example *"company shelves wind sector plans"*.

There are many ways in which social network data can be leveraged to give a better understanding of user opinion such problems are at the heart of natural language processing (NLP) and data mining research.

In this paper we present a tool for sentiment analysis which is able to analyse Twitter data. We show how to automatically collect a corpus for sentiment analysis and opinion mining purposes. Using the corpus we build a sentiment classifier, that is able to determine positive, negative and objective sentiments for a document.

1.1 Related Work

The increase in social media networks has made sentiment analysis a popular research area, in recent years. In Turney[4] reviews are classified by calculating the summation of polarity of the adjectives and adverbs contained within text. This study utilised movie and car reviews, where thumbs up and thumbs down ratings indicate positive and negative sentiment respectively. A discrepancy between the accuracy of the movie and car reviews was observed with the car reviews getting a higher accuracy. This was attributed to the fact that movie reviews, whilst being positive, can have a lot of adjectives and adverbs that do not fully relate to the overall enjoyment of the film and can actually be more a description of the scenes within the film itself. The PMI-IR (Pointwise Mutual Information - Informations Retrieval) algorithm was used to classify documents. This algorithm works by taking the relevant bigrams from the document then using the near function on a search engine to see how many times this bigram appears near a word that expresses strong positive or negative sentiment, a large number of matches indicates a stronger polarity.

Pang[1] consider word presence vs frequency where word presence is found to be more effective than word frequency for sentiment analysis. Word position within a given sentence can also be effective, where such information can be used to decide if a particular word has more strength at the beginning or the end of a given sentence.

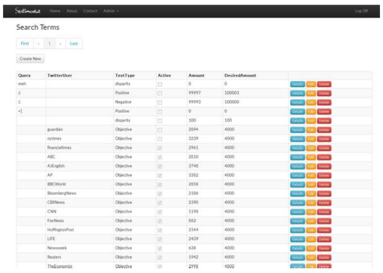
Go[2] train sentiment classifier on Twitter data. This itself presents a new challenge as there is no explicit rating system such as star rating or thumbs rating like in previous work. This issue is negated through the use of Twitter's search functionality by searching for emoticons such as :) :(representing positive and negative sentiment respectively. This system is highly limited as it is restricted to binary classification and does not take into account objective texts. This work explored the use of several different classifiers across different n-grams with and without the use of POS tags. A combination of using Unigrams and Bigrams give the best results across all classifiers. The inclusion of POS tags with unigrams had a negative impact across all classifiers however this still performed better than using bigrams. Our work considers combination of Bigrams and POS tags.

Pak[3] considers objective tweets as well as those that are positive and negative sentiment. This paper discuses the method for collecting corpus data, this again is similar to [2] by using emoticons for positive and negative sets. As it is also concerned with collecting data for an objective set it looks at the tweets from a collection of news sources such as the New York Times, Financial Times etc. Pak[3]. provide a rigorous analysis of their corpus, showing sets of texts differ in terms of the POS tag distributions. Generally there is a far greater difference in the objective and subjective texts than positive and negative sets, such differences show that using POS tags can be a strong indicator of the difference between types of text. The objective and subjective comparison shows that the interjections and personal pronouns are strong indicators of subjective texts whilst common and proper nouns are indicators of objective texts. Subjective texts are often written in first or second person in past tense whilst objective texts are often written in third person. The difference between the positive and negative sets do not give a strong indication, however they are good indicators in the difference between the amount of superlative adverbs and possessive endings, both indicating positive sentiment whilst the negative set often contains more verbs in the past tense as people are often expressing disappointment.

Pak[3] use multi nominal naive Bayes classifier to compare unigrams, bigrams and trigrams they conclude that bigrams give the best coverage in terms of context and expression of sentiment. Pak[3] also compare the usage of using negation attachment to words although this process may be considered unorthodox it does improve the classification process by 2% on average. Pak[3] also consider use of two different methods for reducing the influence of words which occurrence is ambiguous between sets, entropy and salience, out of these two salience was found to work better however the use of these methods can introduce ambiguity into the system meaning that the classification process may fail depending on the filter value selected. The simplification of their calculation for classification is potentially dangerous as this assumes there is equal word distribution across sets, having run this test on our data set we have found that the negative set contained over 4% more words than the positive set, showing clear bias in the classification process. The method used for reporting accuracy, is through the process of plotting accuracy against decision. This essentially allows the system to cherry pick data and claim high accuracy across a small subsection of the testing data whilst ignoring the rest.

2 Sentimator: Sentiment Analysis Tool

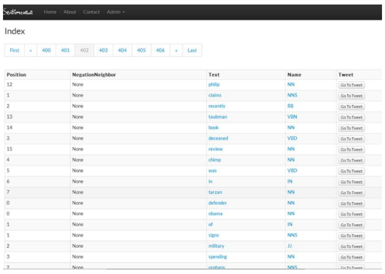
Sentimator[1] is a web based tool which uses naive Bayes Classifier to classify live Twitter data based on positivity, negativity and objectivity. Sentimentor has an interface which enables the user to analyse the word distributions(see Figures 1 and 2. Sentimentor presents classification results in a easy to understand pictorial format (see Figure 3). Other functionalities of Sentimentor include: the text type details(see Figure 4); The analysis of the twitter message (see Figure 5); search (see Figure 6).



Search Terms

Query	TwitterUser	TextType	Active	Amount	DesiredAmount	
meh		disparity	☐	0	0	Details Edit Delete
:)		Positive	☐	99997	100003	Details Edit Delete
:(		Negative	☐	99993	100000	Details Edit Delete
=]		Positive	☐	0	0	Details Edit Delete
		disparity	☐	100	100	Details Edit Delete
	guardian	Objective	☐	2894	4000	Details Edit Delete
	nytimes	Objective	☐	3239	4000	Details Edit Delete
	financialtimes	Objective	☑	2961	4000	Details Edit Delete
	ABC	Objective	☑	2810	4000	Details Edit Delete
	AJEnglish	Objective	☑	3748	4000	Details Edit Delete
	AP	Objective	☑	3382	4000	Details Edit Delete
	BBCWorld	Objective	☑	2858	4000	Details Edit Delete
	BloombergNews	Objective	☑	2186	4000	Details Edit Delete
	CBSNews	Objective	☑	2390	4000	Details Edit Delete
	CNN	Objective	☑	1198	4000	Details Edit Delete
	FoxNews	Objective	☑	882	4000	Details Edit Delete
	HuffingtonPost	Objective	☑	2344	4000	Details Edit Delete
	LIFE	Objective	☑	2439	4000	Details Edit Delete
	Newsweek	Objective	☑	638	4000	Details Edit Delete
	Reuters	Objective	☑	1942	4000	Details Edit Delete
	TheEconomist	Objective	☑	2998	4000	Details Edit Delete

Fig. 1. Screenshot of the search term index

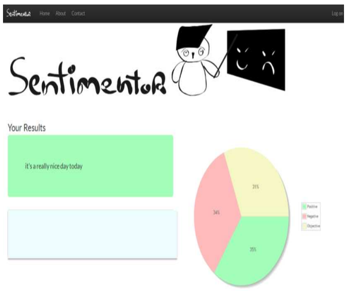


Index

Position	NegationNeighbor	Text	Name	Tweet
12	None	philip	NN	Go To Tweet
1	None	claims	NNS	Go To Tweet
2	None	recently	RB	Go To Tweet
13	None	taubman	VBN	Go To Tweet
14	None	book	NN	Go To Tweet
3	None	deceased	VBD	Go To Tweet
15	None	review	NN	Go To Tweet
4	None	chimp	NN	Go To Tweet
5	None	was	VBD	Go To Tweet
6	None	in	IN	Go To Tweet
7	None	tarzan	NN	Go To Tweet
0	None	defender	NN	Go To Tweet
0	None	obama	NN	Go To Tweet
1	None	of	IN	Go To Tweet
1	None	signs	NNS	Go To Tweet
2	None	military	JJ	Go To Tweet
3	None	spending	NN	Go To Tweet
2	None	orphans	NNS	Go To Tweet

Fig. 2. Screenshot of the word position index

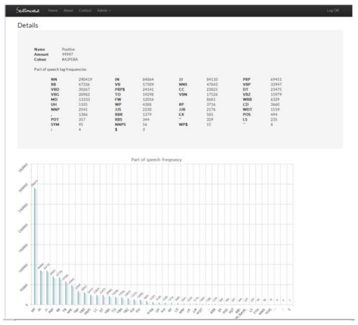
[1] http://sentimentor.co.uk



Your Results

it's a really nice day today

Fig. 3. Sentiment analysis of a piece of text



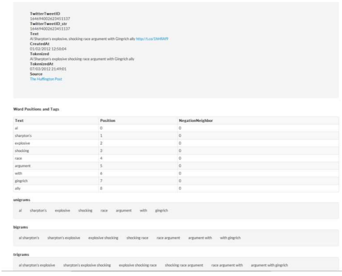
Details

Name Positive
Amount 99997
Colour #A3FEBA

Part of speech tag frequencies

NN	290419	**IN**	84864	**JJ**	84110	**PRP**	69451
RB	67336	**VB**	57509	**NNS**	47842	**VBP**	33947
VBD	30267	**PRP$**	24141	**CC**	23825	**DT**	23475
VBG	20982	**TO**	19298	**VBN**	17526	**VBZ**	15979
MD	13333	**FW**	12056		8681	**WRB**	6329
UH	5105	**WP**	4388	**RP**	3716	**CD**	3660
NNP	2541	**JJS**	2230	**JJR**	2176	**WDT**	1559
,	1386	**RBR**	1379	**EX**	585	**POS**	494
PDT	357	**RBS**	344	**``**	329	**LS**	235
SYM	95	**NNPS**	56	**WP$**	15	**''**	8
:	4	**$**	2				

Fig. 4. Screenshot of the text type details



TwitterTweetID
164694002623451137
TwitterTweetID_str
164694002623451137
Text
Al Sharpton's explosive, shocking race argument with Gingrich ally http://t.co/1hH6Af9
CreatedAt
01/02/2012 12:58:04
Tokenized
Al Sharpton's explosive shocking race argument with Gingrich ally
TokenizedAt
07/03/2012 21:49:01
Source
The Huffington Post

Word Positions and Tags

Text	Position	NegationNeighbor
al	0	0
sharpton's	1	0
explosive	2	0
shocking	3	0
race	4	0
argument	5	0
with	6	0
gingrich	7	0
ally	8	0

unigrams

al sharpton's explosive shocking race argument with gingrich

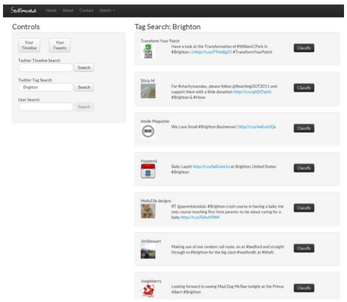
bigrams

al sharpton's sharpton's explosive explosive shocking shocking race race argument argument with with gingrich

trigrams

al sharpton's explosive sharpton's explosive shocking explosive shocking race shocking race argument race argument with argument with gingrich

Fig. 5. Screenshot of the tweet details page



Controls

Twitter Timeline Search

Twitter Tag Search
Brighton

User Search

Tag Search: Brighton

Transform Your Patch
Have a look at the Transformation of #WilliamCPark in #Brighton :) http://t.co/FYUo8g55 #TransformYourPatch

Silvia M
For #charitytuesday, please follow @ReachingOUT2011 and support them with a little donation: http://t.co/ghDiTqmh #Brighton & #Hove

Inside Magazine
We Love Small #Brighton Businesses! http://t.co/lwEsxUQx

Happenic
Baby Lapsit http://t.co/bdGore1e at Brighton, United States #Brighton

MollyElla designs
RT @parentskooluk: #Brighton crash course in having a baby the only course teaching first time parents-to-be about caring for a baby http://t.co/Fj4xAPW4

JimStewart
Making use of one random rail route, on at #bedford and straight through to #brighton for the big clash #watfordfc at #bhafc

Joogleberry
Looking forward to seeing Mad Dog McRae tonight at the Prince Albert #Brighton

Fig. 6. Screenshot of the twitter search functionality

3 Data Collection and Preprocessing

Twitter API was used for the data extraction process. Negative, positive and objective texts were collected by following the same procedures as in([2] and [3]). Tokenization process from [2] and [3] was followed for the data preprocessing task. The steps followed included the removal of any urls and usernames (usernames follow the @symbol) and removal any characters that repeat more than twice turning a phrase such as OOMMMGGG to OOMMGG, which is applied by a regular expression. Table 1, shows an example of the tokenization process. Finally, the stopset words were removed from the data. The stopset is the set of words such as “a”, “and”, “an”, “the”, these are words that do not have any meaning on their own. The second phase is associated with determining the POS tag for each word. The OpenNLP library was used for POS tagging and the extraction of unigrams and bigrams. An example of bigrams extracted from our dataset is shown in Tables 2, 3 and 4.

Table 1. Example of the Tokenization Process

	Before	After
1	I wanna go to @AvrilLavigne 's concert stadium merdeka sooooooooo badly :(love you avril! Xo	in I wanna go to s concert in stadium merdeka soo badly love you avril Xo
2	@chuckcomeau 1:45am!!! OMG I WAS SLEPT AT 11:00pm WOOOOOOW I WANT A SKATE :)	am OMG I WAS SLEPT AT pm WOOW I WANT A SKATE
3	British adventurer Felicity Aston becomes first woman to ski across Antarctica	British adventurer Felicity Aston becomes comes first woman to ski across Antarctica alone

Table 2. Positive Bigram Counts

Bigram	Count
i love	2899
valentines day	2797
happy valentines	2191
thank you	2141
love you	2133
follow back	1516
d rt	1491
think i'm	1410
follow me	1342
if you	1263

Table 3. Negative Bigram Counts

Bigram	Count
i miss	3292
i have	2440
i don't	2041
i was	1922
i want	1881
but i	1813
i know	1760
miss you	1681
want to	1609
i can't	1595

Table 4. Objective Bigram Counts

Bigram	Count
to be	916
front page	574
new york	524
if you	506
in today's	496
out of	430
will be	426
mitt romney	418
us your	397
more than	395

3.1 Evaluation of the data set

An original corpus of Twitter data was collected and compared with the corpus presented in Pak[3]. The percentage distribution of POS tags across sets is shown in Figure 7.

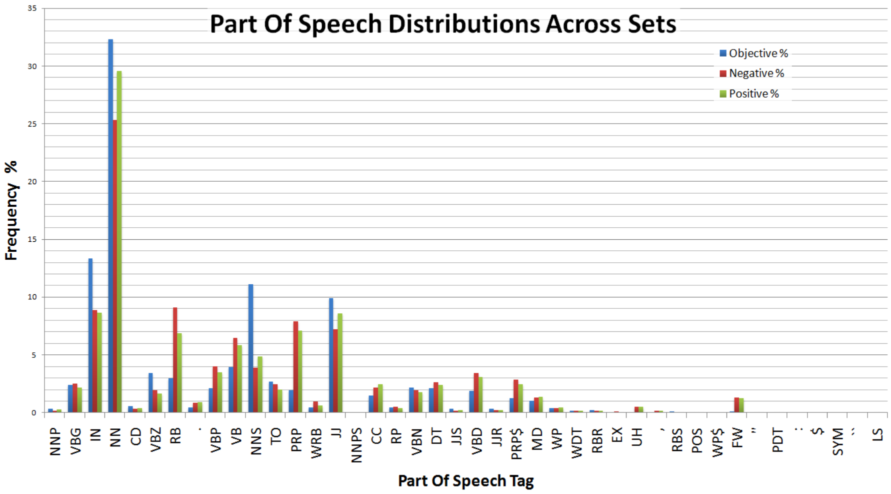


Fig. 7. The percentage distribution of POS tags across the sets

Overall singular noun (NN) is the most common POS tag, occurring 29.08% across the whole corpus. Preposition or conjunction (IN) occur 10.28% of the time with it being clear that there is a significant difference between the occurrence in all sets. To better understand the differences between sets we have calculated the percentage difference between the percentage distribution of each POS tag. This has been done for the difference between the objective and subjective sets and between the positive and negative sets, this is displayed in Figure 8 and Figure 9 respectively. Figure 8 shows a significant difference in the amount of interjections (UH) and personal pronouns (PRP, PRP$) favouring the subjective set as reported by Pak[3] . The common nouns and proper nouns are a strong indicator of the subjective set by looking at common noun plural (NNS) nouns proper singular (NNP) and noun common singular (NN). According to Pak[3] we expect writers of subjective text to be talking in the first or second person, we can partially confirm this by looking at the difference of verb present tense not third person singular (VBP) and verb past tense (VPD) however verb, present participle (VBG) contradicts this as it prevails in the objective set. This could have happened because the selected news outlets might have more comment on news than original reporting or this could be a difference in the POS tagger, however this is of little concern because the difference is relatively negligible. Likewise we can expect objective texts to be in third person the results for

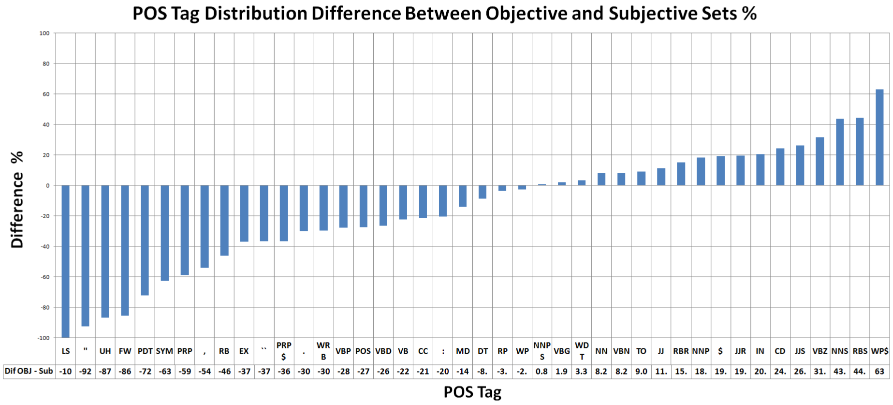


Fig. 8. Graph showing the percentage difference of POS tags frequencies between objective and subjective texts

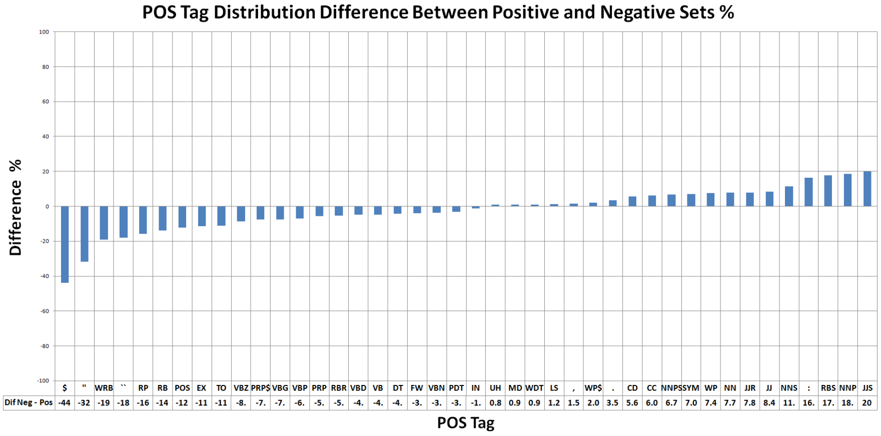


Fig. 9. Graph showing the percentage difference of POS tag frequencies between positive and negative sets

verb present tense 3rd person singular (VBZ) can confirm this. In our dataset Superlative adverbs (RBS) have a very Strong weighting for objective text this is contrary to Pak[3] , where this is not significant [16]. The List item marker (LS) has a -100% difference as this doesn't occur in the objective set this tag isn't present in Pak [3] data. The symbols that have not been removed by the tokenizer are a potential source of error as these represent significant difference between sets. The POS tagger used has the ability to detect foreign words (FW) which have a Strong indication of the text being subjective, the reasoning for this is because news outlets would only be expected to use correctly structured English, standard user tweets may contain a mix of languages despite the fact that the Twitter search was limited to English tweets. Now looking at Figure 9 we can see that these two sets are a lot closer in terms of POS difference, which is expected as both sets are subjective. The strongest indicators for negative sentiment is Currency ($) and quotation marks while an individual is highly likely to express their fiscal issues in a negative sentiment but as there are only 19 occurrences of currency in the system this is not a good indicator of what set the text belongs to, also the inclusion of quotation marks here is likely going to introduce error into the system. Wh-adverb - negative (WRB) , particle (RB, RP) genitive marker (POS) are all strong indicators on negative sentiment, however Pak[3]. state that (POS) may be an indicator of positive sentiment, the results we have collected contradict this. Superlative adverb (RBS) , proper noun singular (NNP) , adjective superlative (JJS) , Noun (NNS) common plural are all indicating strong positive sentiment. The appearance of (RBS) confirms that this is a good indicator of a positive sentiment.

3.2 Classification

The naive Bayes classifier was used for classification this decision is primarily based on findings by Pak and Go [[2],[3]], that the naive Bayes classifier show good performance results.

$$P(C|m) = P(C)\prod_{i=1}^{n} P(f_i|C)$$

where C is the class positive, negative or objective sets, m is the twitter message and f is a feature. In our experiments the features are POS tags, unigrams or bigrams.

4 Results

We have tested our classifier against a training set which contains 216 manually tagged tweets. We have provided the test results for unigrams and bigrams both with and without the use of POS tags these results are detailed in Tables 5,6,7, 8 and 9. Table 9 details the accuracy of each of the previously mentioned tests. The test with the highest accuracy is the one using bigrams without POS tags with an accuracy of 52.31% and the lowest is Unigrams without POS tags at 46.76%. Accuracy would be far higher if we were to carry out these tests using binary classification and it should be stated that this is one of the further complexities of using microblogging data as appose to using reviews as these are not expected to be objective. The use of bigrams has show an increase in performance with or without the use of POS tags. This also reduces the amount of false positives in the objectivity classifier however there is also notable increase in false positives by the positive classifier, the negative classifier does not seem to be effected much by this. Overall the use of POS tags has had a negative effect on the accuracy of the calssification proccess, this is caused by the Ambiguity of POS tag occurances across sets this is most likely also the case because we using the summation of POS tags in a given phrase and not looking for binary occurance as disscused in [1]. It may potentially benifit the classifcation proccess to give less wheight to the POS tags or to experiment with diffrent n-grams of POS tags. We have confirmed previous works finding to be correct in there conclusion that bigrams give better results than unigrams. The overall performance of the system is satisfactory, however we would still like to further improve this as outlined in our future work section.

Table 5. Results for Unigrams

Sentiment	Number of Samples	Correctly Identified	False Positives
Positive	108	37	9
Negative	75	45	45
Objective	33	19	61

Table 6. Results for Unigrams and POS Tags

Sentiment	Number of Samples	Correctly Identified	False Positives
Positive	108	39	10
Negative	75	45	42
Objective	33	18	62

Table 7. Results for Bigrams

Sentiment	Number of Samples	Correctly Identified	False Positives
Positive	108	47	16
Negative	75	47	44
Objective	33	19	43

Table 8. Results for Bigrams and POS Tags

Sentiment	Number of Samples	Correctly Identified	False Positives
Positive	108	46	19
Negative	75	45	43
Objective	33	17	46

Table 9. Results compared

Test	Correctly Identified	False Positives
Unigrams	101	46.75%
Unigrams POS	109	47.2%
Bigrams	113	52.31%
Bigrams POS	108	50%

5 Conclusion and Future Work

In this paper we have presented a way in machine learning techniques can be applied to large sets of data to establish membership, in this case positivity, negativity and objectivity. We have looked at common process in NLP that can help us derive the meaning or context of a given phrase. We have demonstrated how to collect an original corpus for sentiment classification and the refinement that is needed with such data. We have applied a naive Bayes classifier to this set conduct sentiment analysis and have found this process to be successful. On analysis of our results we have confirmed that bigrams offer better performance when conducting the classification process supporting Pak[3] results. We has also confirmed some of Pak[3] findings when looking at the differences between the objectivity and subjectivity set, the same can't be be said for the positive and negative sets which prove to be far more ambiguous. We have discovered that collecting data across a short amount of time may be a potential source of error when determining sentiment, this is due to the fact that opinions can shift over time as can the meaning of words. The classification process itself has been successful with and accuracy of 52.31% however it is felt that this could be further improved, this is outlined in future work. One of our future works is to experiment with different classifiers on our dataset. We also intend on developing an application which carries our textual analysis on video games servers analysing what a player is expressing and adjusting the game enviroment accordingly.

References

1. Bo Pang, L.L.: Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval January Volume 2 Issue 1-2, 1–94 (2008), `http://www.cs.`

cornell.edu/home/llee/omsa/omsa.pdf

2. Go, A., Bhayani, R., Huang, L.: Twitter sentiment classification using distant supervision. Processing 150(12), 1–6 (2009), http://www.stanford.edu/~alecmgo/papers/TwitterDistantSupervision09.pdf
3. Pak, A., Paroubek, P.: Twitter as a corpus for sentiment analysis and opinion mining. In: Chair), N.C.C., Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., Rosner, M., Tapias, D. (eds.) Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10). European Language Resources Association (ELRA), Valletta, Malta (may 2010)
4. Turney, P.D.: Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics. pp. 417–424. ACL '02, Association for Computational Linguistics, Stroudsburg, PA, USA (2002), http://dx.doi.org/10.3115/1073083.1073153

Mining for Opinions Across Domains: A Cross-Language Study[⋆]

Anna Kravchenko

Higher School of Economics,
Research and Educational Center of Information Management Technologies
Kirpichnaya ul. 33/4, 105679 Moscow, Russia

Abstract. An important task in opinion mining is detecting subjective expressions in texts and distinguishing them from factual information. High lexicon diversity between different domains excludes the possibility of formulating universal rules that would work for any area of knowledge. In this article we suggest a solution for this problem. We define the features that most opinionated sentences share and propose a cross-language classification of subjective expressions, illustrated by examples in Russian, English and Chinese. We also propose an algorithm based on this classification that generates a set of extraction patterns for any domain from a corpus of untagged texts. The corpus requires no additional preparation except for POS-tagging. The effectiveness of the proposed approach is evaluated for English and Russian on collections of approximately 300 000 sentences each, gathered from three different domains: user reviews on movies, headphones and photo cameras.

Keywords: opinion mining, sentiment analysis

1 Introduction

One of the main challenges of opinion mining is that subjective expressions vary profoundly, depending on the domain. The exact same word or phrase may or nay not be considered opinionated in different contexts. For example, "short battery life" is clearly a negative opinion, and "short article" is simply a literature genre.

There is a current trend to focus only on machine learning techniques as a workaround for this problem, entirely dismissing the underlying linguistic structure, but we strongly believe it is essential to take it into account as well.

It is easy to see that there are properties that subjective sentences share across domains:

- Syntactic structure of subjective expressions is similar and domain-independent,

[⋆] This work was conducted with financial support from the Government of the Russian Federation (Russian Ministry of Science and Education) under contract № 13.G25.31.0096 on "Creating high-tech production of cross-platform systems for processing unstructured information based on open source software to improve management innovation in companies in modern Russia".

– Some subjective words and expressions are domain-independent ("fine", "terrific", "happy"),
– Opinionated sentences usually contain more than one subjective expression and often occur next to each other in texts.

Those properties allow us to find opinionated sentences in the text, using domain-independent words as pivots, and then to extract domain-specific expressions from them, based on the expected syntactic structure of opinion. Those expressions can be further transformed into extraction patterns and used in mining algorithms.

Therefore it allows us to extract a lexicon of opinionated expressions for such diverse areas as, for example, political news, reviews on movies and reviews on cameras.

It is important that no manual tagging of the processed texts is reqired, which minimizes the need for human participation.

2 Types of opinionated sentences

As it has been mentioned earlier, opinionated expressions share syntactic structure between domains, but there are different ways of expressing an opinion. We propose a classification of subjective expressions, based on study of English, Russian and Chinese. Those languages were chosen as the most heterogeneous examples - Russian has free word order and a well-developed morphology, in English the word order is fixed and morphology is significantly less complex, and Chinese also has fixed word order and very poor morphology. It is also important that Chinese is a Non-Indo-European language.

We will use the term **object** to denote the target entity that has been commented on. We will also use the notions of proposition. A proposition is the semantic core of a clause or a sentence that is constant, despite changes in such things as the voice or illocutionary force of the clause.

Subjective expressions can be divided into the following classes:

Class 1 **Explicit opinion:**

Feature or characteristics with evaluative meaning is directly attributed to the object.
It can be expressed with the following types of propositions:

1.a. **Complete proposition, opinion is expressed by a noun phrase.**

Examples:
English:
the situation seems bad, John is an outstanding painter
Russian:
наушники дрянь, Иванов плохой руководитель
earphones garbage, Ivanov bad manager
Chinese (Pinyin):

Ěrj shì fèiwù, Zhang shì ygè huài de jngl.
earphones is garbage, Zhang is single bad of manager

The proposition can be used as a complete sentence, a copula verb is used (zero copula in Russian, "shì" in Chinese).
Not all examples are consistent for better illustration of material.

1.b. **Complete proposition, opinion is expressed by an adjective phrase.**

Examples:
English:
the sound is terrific
Russian:
звук замечательный
sound terrific
Chinese:
Shngyn hěn dà
sound very good

2.a. **Incomplete proposition, opinion is convoluted with the noun phrase. Genitive case.**

Examples:
English:
photo's great quality, great quality of photos
Russian:
высокое качество фотографий.
hight quality-GEN photo-GEN-PL
Chinese:
doesn't occur in Chinese

It is important that the word "quality" itself bears affective connotation in this case. For example, "big TV screen" would be of type 3.

2.b. **Incomplete proposition, opinion is convoluted with the noun phrase pointing to the object. Nominative case.**

Examples:
English:
high quality photos
Russian:
известный преступник Петров
known criminal Petrov
Chinese:
doesn't occur in Chinese

3. **Incomplete proposition, opinion is expressed by adjective.**

   Examples:
   English:
   amazing design
   Russian:
   замечательный дизайн
   amazing design
   Chinese:
   Yuzhì de yìngxiàng
   high quality image

4. **Complete proposition, describing a situation involving the object. Opinion is expressed by a verb phrase.**

   Examples:
   English:
   the phone broke quickly
   Russian:
   телефон быстро сломался
   phone quickly broke-PAST
   Chinese:
   Diànhuà hěn kuài dpòle
   phone very quickly broke

Class 2. **Direct affective connotation.**

Object is characterised by its relation to entities with strong affective connotations. For example, "The President fights against corruption", "people criticize the government".
This type is expressed by a complete proposition, the semantic orientation of opinion is formed by the semantic orientation of the predicate and the associated entity.

Class 3. **Associated affective connotation.**
Object is characterised by a class of situations appearing in the same text or sentence, but not related directly to the object. For example, "Impoverishment risks", "bought a new pair".
This type can be expressed by a proposition of any form and is extremely hard to detect with natural language processing methods.

Analysis of classes 2 and 3 is very complex and requires deep syntactic analysis, for this reason we will only focus on expressions of class 1. We will also show that this is sufficient in most applications.

3 Extracting subjective expressions

A lot of opinionated sentences contain more than one subjective expression, and some of those are may be domain-independent. Therefore, if a sentence contains a domain-independent subjective expression and a new domain-specific one, we can extract the latter, if it matches any of the syntactic patterns that we expect.

Most of the time we can also 'guess' its semantic orientation, using the semantic orientatin of the pivot word. It can act as an additional argument when dealing with ambiguity.

This tagging methods was first proposed by Ellen Rillof [11], though she used shallow parsing instead of syntactic patterns. It requires a large enough corpus to process, but unannotated texts are easy to come by, so even if the classifier can label only 30% of the sentences as subjective, it will still produce a large collection of labeled sentences.

Example (known and new words are highlighted in bold and italic respectively):

These are the **best** closed-back headphones I've heard at this price, *bass is intense*, *highs are not shrill*, *no sound leak*, *comfortable design.*

We can extract the following expressions from the example:
bass is intense, [N, V, Adj], type 1.b
highs are not shrill, [N, V, Adj], type 1.b
no sound leak, [Part, N, N], type 2b
comfortable design, [Adj, N], type 3

Those expressions (we will call them **segments**) then can be further transfomed into lexical patterns. Some segments may contain name of the exact model they are describing, while they can in fact be applied to any other model or it's feature. It is important to replace those names with some universal label.

The method is as follows: Each segment is first converted to a sequence. Each sequence element is a word, which is represented by both the word itself and its POS tag in a set. In the training data, all object features or objects' names are labeled and replaced by the label $feature according to the original segment syntactic structure.
For example, the sentence segment, "Included memory is stingy", is turned into the sequence:
{included, Adv}{memory, N}{is, V}{stingy, Adj}.
After labeling, it becomes an extraction pattern (note that "memory" is an object feature):
{included, Adv}{$feature, N}{is, V}{stingy, Adj},

Feature extraction is performed by matching the patterns with each sentence segment in a new review to extract object features. That is, the word in the sentence segment that matches $feature in a pattern is extracted

A similar method is described by Bing Liu [8].

4 Filtering

Not all extracted patterns will indeed be subjective. Choosing which patterns to keep usually requires a human expert. Riloff also proposed a method for minimizing human participation. The method is based on a tendency of subjective expressions to reappear in multiple subjective sentences in the text more often than the expressions extracted by mistake.

All extraction patterns are ranked using a conditional probability measure: the probability that a sentence is subjective given that a specific extraction pattern appears in it.

The exact formula is:

$$P(subjective/pattern_i) = subjfreq(pattern_i)/freq(pattern_i),$$

where $subjfreq(pattern_i)$ is the frequency of $pattern_i$ in subjective training sentences, and $freq(pattern_i)$ is the frequency of $pattern_i$ in all training sequences.

A thresholds are used to select extraction patterns. We choose extraction patterns for which $pr(subkective/pattern_i) > \theta$. The threshold is chosen manually.

5 Evaluation

Currently the methods of testing sentiment analysis systems are not fully developed. For this reason we use a method based on subjective evaluations of small text collection by an expert.

Expert marks each pattern as evaluative or extracted by mistake, and precision is then calculated using the following formula:

$$P = N_{subj}/N_{all},$$

where N_{subj} is the number of correctly extracted patterns and N_{all} is the number of all extracted patterns. We do not evaluate recall, because of the amount of expert work it requires and because this method does not provide high recall by design, which can be compensated by the corpus size and automatic tagging.

For the evaluation process two corpora of approximately 300,000 sentences each were collected for three languages. All three consisted of three parts - reviews on photo cameras, earphones and movies.

Results show that the algorithm manages to extract opinionated phrases from texts of all three domains, though the accuracy differs. For domains with objective evaluation criteria and relatively low lexical variability (for example, reviews on earphones and photo cameras) shows good precision: 52% before filtering and 80% after filtering for Russian and 67% and 83% acorrdingly for English, with $\theta = 0,9$. For movie reviews precision was much lower, 29% before filtering and 64,3% after filtering for Russian and 31% and 68% for English with $\theta = 0,6$.

Precision can be made higher by increasing θ, but it lowers recall valye and requires a significantly larger corpus,

6 Conclusion

As results show, the proposed method achieves high enough precision for texts on certain subjects and can be further used as a component of opinion extraction system. Presicion can be enhanced by improving the size and quality of the training corpora.

It is important that the method requires almost no manual preparation, and a collection of texts on certain topic can be easily acquired for example by searching for specific category in online stores.

The direction of further work is creating a full opinion mining based on the proposed algorithm and classification.

References

1. Carenini G., Pauls A.: Multi-Document Summarization of Evaluative Text in Proceedings of EACL 2006, Trento, Italy, pp. 305-312. (2006)
2. Ermakov A.E., Kiselev S.L.: Linguistic Model For Computational Sentiment Analysis of Media, Proceedings of the International Conference Dialog 2005, Moscow. (2005)
3. Hatzivassiloglou V., McKeown K.: Predicting the semantic orientation of adjectives in Proceedings of ACL/EACL 1997, Madrid, Spain, Complutense University of Madrid, pp. 174-181. (1997)
4. Hu M. and Liu B.: Mining and Summarizing Customer Reviews in Proceedings of KDD-2004, Seattle, WA, 2004, pp. 168-177. (2004)
5. Hu M. and Liu B.: Mining Opinion features in Customer Reviews in Proceedings of AAAI'04, Boston, Massachusetts, USA: AAAI Press, pp. 755-760. (2004)
6. Kobayashi N., Inui K., Tateishi K., Fukushima T.: Collecting Evaluative Expressions for Opinion Extraction in Proceedings of IJCNLP-2004, Berlin, Germany: Springer, pp. 596-605. (2004)
7. Liu, B.: Web Data Mining, Springer, Berlin. (2007)
8. Liu, B.: Sentiment Analysis and Subjectivity in Handbook of Natural Language Processing, Second Edition, Chapman and Hall/CRC, NY, USA, pp. 257-282. (2010)
9. Nozhov I. (2003) Morphologic and Syntactic Text Processing (Models and Computations), available at http://www.aot.ru/docs/Nozhov/msot.pdf
10. Popescu A.M., Etzioni O.: Product features and Opinions from Reviews in Proceedings of HLT-EMNLP 2005. Vancouver, Canada: ACL, pp. 339-346. (2005)
11. Riloff E, Wiebe J.: Learning Extraction Patterns for Subjective Expressions, in Proceedings of EMNLP-03, Sapporo, Japan, pp. 97-104. (2003)
12. Turney P.: Inference of Semantic Orientation from Association, available at `http://cogprints.org/3164/01/turney-littman-acm.pdf` (2003)
13. Turney P.: Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews in Proceedings of ACL 2002, Philadelphia, PA, U.S.A., pp. 417-424. (2002)

Comparative Experiments for Multilingual Sentiment Analysis Using Machine Translation

Alexandra Balahur and Marco Turchi
alexandra.balahur@jrc.ec.europa.eu
marco.turchi@jrc.ec.europa.eu

European Commission Joint Research Centre
IPSC, GlobeSec, OPTIMA
Via E. Fermi 2749, Ispra, Italy

Abstract. Sentiment analysis is the Natural Language Processing (NLP) task dealing with sentiment detection and classification from text. Given the importance of user-generated contents on the recent Social Web, this task has received much attention from the NLP research community in the past years. Sentiment analysis has been studied in different types of texts and in the context of distinct domains. However, only a small part of the research concentrated on dealing with sentiment analysis for languages other than English, which most of the times lack or have few lexical resources In this context, the present article proposes and evaluates the use of machine translation and supervised methods to deal with sentiment analysis in a multilingual context. Our extensive evaluation scenarios, for German, Spanish and French, using three different machine translation systems and various supervised algorithms show that SMT systems can start to be employed to obtain good quality data for other languages. Subsequently, this data can be employed to train classifiers for sentiment analysis in these languages, reaching performances close to the one obtained for English.

1 Introduction

During the past years, the contents that are generated by users on the Web, in the form of comments and statements of opinions in fora, blogs, reviewing sites, microblogs, have become more and more important. Their high volume and unbiased nature, as well as the fact that they are written by people from all social categories, all over the world, make such information useful to many domains, such as Economics, Social Science, Political Science, Marketing, to mention just a few. Nevertheless, the high quantity of such data and the high rate in which it is produced requires that automatic mechanisms are employed in order to extract valuable knowledge from it. In the case of opinionated data, this issue motivated the rapid and steady growth in interest from the Natural Language Processing (NLP) community to develop computational methods to analyze subjectivity and sentiment in text. These tasks received many names, from which "subjectivity analysis", "sentiment analysis" and "opinion mining" are the most frequently employed ones. The body of research conducted within these tasks has proposed different methods to deal with subjectivity and sentiment classification in different texts and domains, reaching satisfactory levels of performance for English. However, for certain applications, such as news monitoring, the information in languages other than English is also highly relevant and cannot be disregarded, as it represents a high percentage of relevant data. In this type of systems, additionally, sentiment analysis tools must be reliable and perform at similar levels as the ones implemented for English.

In order to overcome the above-mentioned issue, the work presented herein aims to propose and evaluate different methods for multilingual sentiment analysis using machine translation and supervised methods. In particular, we will study this issue in three languages - French, German and Spanish - using three different Machine Translation systems - Google Translate, Bing Translator[1] and Moses [11] and different machine learning models. To have a more precise measure of the impact of quality translation on this task, we create Gold Standard sets for each of the three languages.

Our experiments show that machine translation systems are reaching a reasonable level of maturity so as to be employed for multilingual sentiment analysis and that for some languages (for which the translation quality is high enough) the performance that can be attained is similar to that of systems implemented for English, in terms of weighted F-measure.

2 Related Work

Most of the research in subjectivity and sentiment analysis was done for English. However, there were some authors who developed methods for the mapping of subjectivity lexicons to other languages. To this aim, [9] use a machine translation system and subsequently use a subjectivity analysis system that was developed for English to create subjectivity analysis resources in other languages. [12] propose a method to learn multilingual subjective language via cross-language projections. They use the Opinion Finder lexicon [22] and use two bilingual English-Romanian dictionaries to translate the words in the lexicon. Another approach was proposed by Banea et al. [3]. To this aim, the authors perform three different experiments - translating the annotations of the MPQA corpus, using the automatically translated entries in the Opinion Finder lexicon and the third, validating the data by reversing the direction of translation. In a further approach, Banea et al. [2] apply bootstrapping to build a subjectivity lexicon for Romanian, starting with a set of 60 words which they translate and subsequently filter using a measure of similarity to the original words, based on Latent Semantic Analysis (LSA) [8] scores. Yet another approach to mapping subjectivity lexica to other languages is proposed by Wan (2009), who uses co-training to classify un-annotated Chinese reviews using a corpus of annotated English reviews. [10] create a number of systems consisting of different subsystems, each classifying the subjectivity of texts in a different language. They translate a corpus annotated for subjectivity analysis (MPQA), the subjectivity clues (Opinion Finder) lexicon and re-train a Naive Bayes classifier that is implemented in the Opinion Finder system using the newly generated resources for all the languages considered. [4] translate the MPQA corpus into five other languages (some with a similar ethimology, others with a very different structure). Subsequently, they expand the feature space used in a Naive Bayes classifier using the same data translated to 2 or 3 other languages. Finally, [18, 19] create sentiment dictionaries in other

[1] http://translate.google.it/ and http://www.microsofttranslator.com/

languages using a method called “triangulation”. They translate the data, in parallel, from English and Spanish to other languages and obtain dictionaries from the intersection of these two translations.

Attempts to use machine translation in different natural language processing tasks have not been widely used due to poor quality of translated texts, but recent advances in Machine Translation have motivated such attempts. In Information Retrieval, [17] proposed a comparison between Web searches using monolingual and translated queries. On average, the results show a drop in performance when translated queries are used, but it is quite limited, around 15%. For some language pairs, the average result obtained is around 10% lower than that of a monolingual search while for other pairs, the retrieval performance is clearly lower. In cross-language document summarization, [21, 5] combined the MT quality score with the informativeness score of each sentence in a set of documents to automatically produce summary in a target language using a source language texts. In [21], each sentence of the source document is ranked according both the scores, the summary is extracted and then the selected sentences translated to the target language. Differently, in [5], sentences are first translated, then ranked and selected. Both approaches enhance the readability of the generated summaries without degrading their content.

3 Motivation and Contribution

The work presented herein is mainly motivated by the need to develop sentiment analysis tools for a high number of languages, while minimizing the effort to create linguistic resources for each of these languages in part. Unlike approaches we presented in Related Work section, we employ fully-formed machine translation systems. In this context, another novelty in our approach is that we also study the influence of the difference in translation performance has on the sentiment classification performance.

Additionally, whereas the distinct characteristics of translated data (when compared to the original data) may imply that other features could be more appropriate. Moreover, such approaches have usually employed only simple machine learning algorithms. No attempt has been made to study the use of meta-classifiers to enhance the performance of the classification through the removal of noise in the data.

More specifically, we employ three MT systems - Bing Translator, Google Translate and Moses to translate data from English to three languages - French, German and Spanish. We create a Gold Standard for all the languages, used, on the one hand, to measure the translation quality and to test the performance of sentiment classification on translated (noisy) versus correct data. These correct translations allow us to have a more precise measure of the impact of translation quality on the sentiment classification task. Another contribution this article brings is the study of different types of features that can be employed to build machine learning models for the sentiment task. Further on, apart from studying different features that can be used to represent the training data, we also study the use of meta-classifiers to minimize the effect of noise in the data.

Our comparative results show, on the one hand, that machine translation can be reliably used for multilingual sentiment analysis and, on the other hand, which are the main characteristics of the data for such approaches to be successfully employed.

4 Dataset Presentation and Analysis

For our experiments, we employed the data provided for English in the NTCIR 8 Multilingual Opinion Analysis Task (MOAT)[2]. In this task, the organizers provided the participants with a set of 20 topics (questions) and a set of documents in which sentences relevant to these questions could be found, taken from the New York Times Text (2002-2005) corpus. The documents were given in two different forms, which had to be used correspondingly, depending on the task to which they participated. The first variant contained the documents split into sentences (6165 in total) and had to be used for the task of opinionatedness, relevance and answerness. In the second form, the sentences were also split into opinion units (6223 in total) for the opinion polarity and the opinion holder and target tasks. For each of the sentences, the participants had to provide judgements on the opinionatedness (whether they contained opinions), relevance (whether they are relevant to the topic). For the task of polarity classification, the participants had to employ the dataset containing the sentences that were also split into opinion units (i.e. one sentences could contain two/more opinions, on two/more different targets or from two/more different opinion holders).

For our experiments, we employed the latter representation. From this set, we randomly chose 600 opinion units, to serve as test set. The rest of opinion units will be employed as training set. Subsequently, we employed the Google Translate, Bing Translator and Moses systems to translate, on the one hand, the training set and on the other hand the test set, to French, German and Spanish. Additionally, we employed the Yahoo system (whose performance was the lowest in our initial experiments) to translate only the test set into these three languages. Further on, this translation has been corrected manually by a person, for all the languages. This corrected data serves as Gold Standard[3]. Most of these sentences, however, contained no opinion (were neutral). Due to the fact that the neutral examples are majoritary and can produce a large bias when classifying the polarity of the sentences, we eliminated these examples and employed only the positive and negative sentences in both the training, as well as the test sets. After this elimination, the training set contains 943 examples (333 positive and 610 negative) and the test set and Gold Standard contain 357 examples (107 positive and 250 negative). Although the upper bound for each of the systems would be possible to estimate using Gold Standard for each of the training sets, as well, at this point we considered the scenario that is closer to real situations, in which the issue is related to the inexistence of training data for a specific language.

5 Using Machine Translation for Multilingual Sentiment Analysis

The issue of extracting and classifying sentiment in text has been approached using different methods, depending on the type of text, the domain and the language considered. Broadly speaking, the methods employed can be classified into unsupervised

[2] http://research.nii.ac.jp/ntcir/ntcir-ws8/permission/ntcir8xinhua-nyt-moat.html

[3] We translated the whole sentences, not opinion units separately, so sentences containing multiple opinion units were translated twice. After duplicate elimination, we remained with 400 sentences in the test and Gold Standard sets and 5700 sentences in the training set.

(knowledge-based), supervised and semi-supervised methods. The first usually employ lexica or dictionaries of words with associated polarities (and values - e.g. 1, -1) and a set of rules to compute the final result. The second category of approaches employ statistical methods to learn classification models from training data, based on which the test data is then classified. Finally, semi-supervised methods employ knowledge-based approaches to classify an initial set of examples, after which they use different machine learning methods to bootstrap new training examples, which they subsequently use with supervised methods.

The main issue with the first approach is that obtaining large-enough lexica to deal with the variability of language is very expensive (if it is done manually) and generally not reliable (if it is done automatically). Additionally, the main problem of such approaches is that words outside contexts are highly ambiguous. Semi-supervised approaches, on the other hand, highly depend on the performance of the initial set of examples that is classified. If we are to employ machine translation, the errors in translating this small initial set would have a high negative impact on the subsequently learned examples. The challenge of using statistical methods is that they require training data (e.g. annotated corpora) and that this data must be reliable (i.e. not contain mistakes or "noise"). The lower the performance in classifying, the more sparse will be the feature vectors employed in the machine learning models. However, the larger this dataset is, the less influence the translation errors have.

Since we want to study whether machine translation can be employed to perform sentiment analysis for different languages, we employed statistical methods in our experiments. More specifically, we used Support Vector Machines Sequential Minimal Optimization (SVM SMO), with different types of features (n-grams, presence of sentiment words), since the literature in the field has confirmed it as the best-performing machine learning algorithm for this task [16].

For the purpose of our experiments, three different SMT systems were used to translate the human annotated sentences: two existing online services such as *Google Translate* and *Bing Translator*[4] and an instance of the open source phrase-based statistical machine translation toolkit Moses [11], trained on freely available corpora.This results in 2.7 million sentence pairs for English-French, 3.8 for German and 4.1 for Spanish. All the modes are optimized running the MERT algorithm [13] on the development part of the training data. The translated sentences are recased and detokonized (for more details on the system, please see [20].

6 Experiments

In order to test the performance of sentiment classification when using translated data, we employed supervised learning using Support Vector Machines Sequential Minimal Optimization [14] - SVM SMO - with different features:

- In the first approach, we represented, for each of the languages and translation systems, the sentences as vectors, whose features marked the presence/absence

[4] http://translate.google.com/ and http://www.microsofttranslator.com/

(boolean) of the unigrams contained in the corresponding training set (e.g. we obtained the unigrams in all the sentences in the training set obtained by translating the English training data to Spanish using Google and subsequently represented each sentence in this training set, as well as the test set obtained by translating the test data in English to Spanish using Google marking the presence of the unigram features).

- In the second approach, we represented the training and test sets as in the previous representation, with the difference that the features were computed not as the presence of the unigrams, but the tf-idf score of that unigram.
- In the third approach, we represented, for each of the languages and translation systems, the sentences as vectors, whose features marked the presence/absence of the unigrams and bigrams contained in the corresponding training set.

In our experiments, we also studied the possibility to employ sentiment-bearing words in the sentences to be classified as features for the machine learning algorithm. In order to do this, we employed the SentiWordNet, General Inquirer and WordNet Affect dictionaries for English and the multilingual dictionaries created by (Steinberger et al., 2012). The main problem of this approach was, however, that very few features were found, for a small number of the sentences to be classified, on the one hand because affect is not expressed in these sentences using lexical clues and, on the other hand, because the dictionaries we had at our disposal for languages other than English were not very large (around 1500 words). For this reason, we will not report these results.

Table 1 presents the number of unigram and bigram features employed in each of the cases.

Language	SMT system	Nr. of unigrams	Nr. of bigrams
	—	5498	15981
English French	Bing	7441	17870
	Google	7540	18448
	Moses	6938	18814
	Bing+Google+Moses	9082	40977
German	Bing	7817	16216
	Google	7900	16078
	Moses	7429	16078
	Bing+Google+Moses	9371	36556
Spanish	Bing	7388	17579
	Google	7803	18895
	Moses	7528	18354
	Bing+Google+Moses	8993	39034

Table 1. Features employed for representing the sentences in the training and test sets.

Subsequently, we performed two sets of experiments:

- In the first set of experiments, we trained an SVM SMO classifier on the training data obtained for each language, with each of the three machine translations, separately (i.e. we generated a model for each of the languages considered, for each of the machine translation systems employed), using the three types of aforementioned features. Subsequently, we tested the models thus obtained on the corresponding test set (e.g. training on the Spanish training set obtained using Google Translate and testing on the Spanish test set obtained using Google Translate) and on the Gold Standard for the corresponding language (e.g. training on the Spanish training set obtained using Google Translate and testing on the Spanish Gold Standard). Additionally, in order to study the manner in which the noise in the training data can be removed, we employed one meta-classifier - Bagging [6] (with varying sizes of the bag and SMO as classifier). In related experiments, we also employed other meta-classifiers, such as AdaBoost[1]), but the best results were obtained using Bagging.
- In the second set of experiments, we combined the translated data from all three machine translation systems for the same language and created separate models based on the three types of features we extracted from this data (e.g. we created a Spanish training model using the unigrams and bigrams present in the training sets generated by the translation of the training set to Spanish by Google Translate, Bing Translator and Moses). We subsequently tested the performance of the sentiment classification using the Gold Standard for the corresponding language, represented using the corresponding set of features of this model.

The results of the experiments (in terms of weighted F-score, per language) are presented in Tables 2, 3, 4 and 5, and for the second set of experiments are presented in Table 6.

Feature Representation	Test Set	SMO	Bagging
Unigram	GS	0.683	**0.687**
Unigram tf-idf	GS	0.651	**0.681**
Unigram+Bigram	GS	0.685	**0.686**

Table 2. Results obtained for English using the different representations.

7 Results and Discussion

Generally speaking, from our experiments using SVM, we could see that incorrect translations imply an increment of the features, sparseness and more difficulties in identifying a hyperplane which separates the positive and negative examples in the training phase. Therefore, a low quality of the translation leads to a drop in performance, as the features extracted are not informative enough to allow for the classifier to learn. For German, an agglutinative language, wrong translation also leads to an explosion of features, of which many are irrelevant for the learning process.

Feature Representation	SMT	Test Set	SMO	AdaBoost M1	Bagging	BLEU Score
Unigram	Bing	GS	0.655	0.62	**0.658**	
		Tr	0.655	0.625	**0.666**	0.227
Unigram	Google T.	GS	0.64	0.622	**0.655**	
		Tr	**0.695**	0.645	0.693	0.209
Unigram	Moses	GS	0.649	0.641	**0.675**	
		Tr	**0.666**	0.654	0.661	0.17
Unigram tf-idf	Bing	GS	0.627	0.628	**0.64**	
		Tr	0.654	0.625	**0.673**	0.227
Unigram tf-idf	Google T.	GS	0.626	0.598	**0.643**	
		Tr	0.667	0.627	**0.693**	0.209
Unigram tf-idf	Moses	GS	0.654	0.646	**0.659**	
		Tr	0.664	0.66	**0.673**	0.17
Unigram+Bigram	Bing	GS	0.641	0.631	**0.648**	
		Tr	0.658	0.636	**0.662**	0.227
Unigram+Bigram	Google T.	GS	0.646	0.623	**0.674**	
		Tr	**0.687**	0.645	0.661	0.209
Unigram+Bigram	Moses	GS	0.644	0.644	**0.676**	
		Tr	0.667	0.667	**0.674**	0.17

Table 3. Results obtained for German using the different feature representations.

From Tables 2,3, 4 and 5, we can see that there is a small difference between performances of the sentiment analysis system using the English and translated data, respectively. In the worst case, there is a maximum drop of 12 percentages using SMO and 8 percentages using Bagging. Ideally, to better measure this drop we would have had to use gold standard training data for each language. As mentioned in Section 4, the creation of the gold standard is a very difficult and time consuming task. We are considering the manual translation of the training data into French, German and Spanish for the future work. Nonetheless, the scenario considered was aimed at studying the use of MT for SA in the real-life scenario, in which there is no annotated data for the language on which SA is done.

The noise in the data appears from two sources - namely the incorrect translations or the features that are not appropriate. Manual inspection of the results has shown that in case of German, the tf-idf obtains the best results because it removes irrelevant features (words that are mentioned very few times). On the other hand, for languages for which the translation quality is higher - i.e. Spanish and French in our case - we obtained better results when using a combination of unigrams and bigrams. After manually inspecting the data, we noticed that cleaner are the data the most useful is the unigram and bigram representation, as this representation increases the quantity of useful features for training. This is not the case for German, where this representation increases to a higher degree the noise (the number of noisy features).

In the line of the previous consideration, Bagging, by reducing the variance in the estimated models, produces a positive effect on the performance increasing the F-score, as compared to the learning process and features without Bagging. These improve-

Feature Representation	SMT	Test Set	SMO	AdaBoost M1	Bagging	BLEU Score
Unigram	Bing	GS	0.627	0.62	**0.633**	
		Tr	**0.634**	0.629	0.618	0.316
Unigram	Google T.	GS	0.635	0.635	**0.659**	
		Tr	0.63	0.63	**0.665**	0.341
Unigram	Moses	GS	**0.644**	0.644	0.639	
		Tr	0.675	0.675	**0.676**	0.298
Unigram tf-idf	Bing	GS	**0.659**	0.649	0.655	
		Tr	0.622	0.637	**0.646**	0.316
Unigram tf-idf	Google T.	GS	0.652	0.652	**0.673**	
		Tr	0.624	0.624	**0.637**	0.341
Unigram tf-idf	Moses	GS	0.646	0.646	**0.66**	
		Tr	**0.677**	0.677	0.676	0.298
Unigram+Bigram	Bing	GS	**0.656**	0.658	0.646	
		Tr	0.633	0.633	**0.633**	0.316
Unigram+Bigram	Google T.	GS	0.653	0.653	**0.665**	
		Tr	0.636	0.667	**0.665**	0.341
Unigram+Bigram	Moses	GS	0.664	0.664	**0.671**	
		Tr	0.649	0.649	**0.663**	0.298

Table 4. Results obtained for Spanish using the different feature representations.

ments are larger using the German data, because the poor quality of the its translations increases the variance in the data. For the same reason, Bagging is quite effective when unigrams and bigrams are used to represent low quality translated data. In this work we pair Bagging with SMO, but we are interested in running experiments using weak classifiers such as Naive Bayes or neural networks.

Finally, as expected, the performance of the classification is much higher for data obtained using the same translator than on the Gold Standard. This is true, as the same incorrect translations are repeated in both sets and therefore the learning is not influenced by these mistakes.

Looking at the results in Table 6, we can see that adding all the translated training data together makes the features in the representation more sparse and increases the noise level in the training data, creating harmful effects in terms of classification performance: each classifier loses its discriminative capability. This is not the case when using tf-idf on unigrams, in which case the combination of the data improves the classification, as this type of features deter sparsity in data.

At language level, clearly the results depend on the translation performance. Only for Spanish (for which we have the highest Bleu score), each classifies is able to properly learn from the training data and try to properly assign the test samples. For the other languages, translated data are so noisy that or the classifier is not able to properly learn the correct information for the positive and the negative classes, and this results in the assignment of most of the test points to one class and zero to the other, or there is significant drop in performance, e.g. for the French language, but the classifier is still able to assign the test points to both the classes.

Feature Representation	SMT	Test Set	SMO	AdaBoost M1	Bagging	Bleu Score
Unigram	Bing	GS	0.604	0.634	**0.644**	
		Tr	0.649	0.654	**0.657**	0.243
Unigram	Google T.	GS	0.628	0.628	**0.638**	
		Tr	0.652	0.652	**0.679**	0.274
Unigram	Moses	GS	0.646	**0.666**	0.642	
		Tr	**0.663**	0.657	0.66	0.227
Unigram tf-idf	Bing	GS	**0.646**	0.641	0.645	
		Tr	0.652	0.661	**0.664**	0.243
Unigram tf-idf	Google T.	GS	0.635	0.635	**0.645**	
		Tr	0.672	0.672	**0.68**	0.274
Unigram tf-idf	Moses	GS	**0.656**	0.635	0.653	
		Tr	**0.686**	0.646	0.671	0.227
Unigram+Bigram	Bing	GS	0.644	0.645	**0.664**	
		Tr	0.644	0.649	**0.652**	0.243
Unigram+Bigram	Google T.	GS	0.64	0.64	**0.659**	
		Tr	0.652	0.652	**0.678**	0.274
Unigram+Bigram	Moses	GS	0.633	0.633	**0.645**	
		Tr	0.666	0.666	**0.674**	0.227

Table 5. Results obtained for French using the different feature representations.

The results confirm the capability of Bagging to reduce the model variance and increase the performance in classification, in particular for the ungrams plus tfidf representation or for the Spanish language. In both the cases, performances are really close (for some configurations even better) to what we obtained using each dataset independently.

8 Conclusions and Future Work

The main objective of this work was to study the manner in which sentiment analysis can be done for languages other than English by employing MT systems and supervised learning. Overall, we could see that MT systems have reached a reasonable level of maturity to produce sufficiently reliable training data for languages other than English. Additionally, for some languages, the quality of the translated data is high enough to obtain performances similar to that for the original data using supervised learning without any subsequent meta-classification for noise reduction. Finally, even in the worst cases, when the quality of the translated data is not very high, the drop in performance is of maximum 12% and it can be improved on using meta-classifiers. From the different feature representations, we could see that wrong translations lead to a large number of features, sparseness and noise in the data points in the classification task. This is especially visible in the boolean representation, which is also more sensitive to noise. Through the different types of features and classifiers, we used showing that using unigrams or tf-idf on unigrams as features, and/or Bagging as a meta-classifier, has a

	Unigrams			Unigrams + tfidf			Unigrams+Bigrams		
Language	SMO	AdaBoost M1	Bagging	SMO	AdaBoost M1	Bagging	SMO	AdaBoost M1	Bagging
To German	0.565*	0.563	0.563*	0.658	0.64	**0.665**	0.565*	0.563*	0.565*
To Spanish	0.587	0.599	0.598	0.657	0.646	**0.666**	0.419	0.494	0.511
To French	0.609	0.575	0.578	0.626	0.634	**0.635**	0.25	0.255	0.23

Table 6. For each language, each classifier has been trained merging the translated data coming form different SMT systems, and tested using the Gold Standard. *Classifier is not able to discriminate between positive and negative classes, and assigns most of the test points to one class, and zero to the other.

positive impact in the results. Furthermore, in case of good translation quality, we noticed that the union of the same training data translated with various systems can help the classifiers to learn different linguistic aspects from the same data.

In future work, we plan to further study methods to improve the classification performance, both by enriching the features employed, as well as extending the use of meta-classifiers to enhance noise reduction. In particular, the first step will be to adding specialized features corresponding to words belonging to sentiment lexica (in conjunction to the types of features we have already employed) and include high level syntax information can reduce the impact of the translation errors. Finally, we plan to employ confidence estimation mechanisms to filter the best translations, which can subsequently be employed more reliably for system training.

Acknowledgements

The authors would like to thank Ivano Azzini, from the BriLeMa Artificial Intelligence Studies, for the advice and support on using meta-classifiers. We would also like to thank the reviewers for their useful comments and suggestions on the paper.

References

1. Balahur, A. and Turchi, M. 2012. Multilingual Sentiment Analysis using Machine Translation?. Proceedings of the 3rd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis Workshop, 52 Jeju, Republic of Korea.
2. Banea, C., Mihalcea, R., and Wiebe, J. 2008. A bootstrapping method for building subjectivity lexicons for languages with scarce resources. Proceedings of the Conference on Language Resources and Evaluations (LREC 2008), Maraakesh, Marocco.
3. Banea, C., Mihalcea, R., Wiebe, J., and Hassan, S. 2008. Multilingual subjectivity analysis using machine translation. Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2008), 127-135, Honolulu, Hawaii.
4. Banea, C., Mihalcea, R. and Wiebe, J. 2010. Multilingual subjectivity: are more languages better?. Proceedings of the International Conference on Computational Linguistics (COLING 2010), p. 28-36, Beijing, China.
5. Boudin, F. and Huet, S. and Torres-Moreno, J.M. and Torres-Moreno, J.M. 2010. A Graph-based Approach to Cross-language Multi-document Summarization. Research journal on Computer science and computer engineering with applications (Polibits), 43:113–118.

6. Breiman, L 1996. Bagging predictors. Machine learning, 24(2):123–140.
7. P. F. Brown, S. Della Pietra, V. J. Della Pietra and R. L. Mercer. 1994. The Mathematics of Statistical Machine Translation: Parameter Estimation. Computational Linguistics 19:263–311.
8. Deerwester, S., Dumais, S., Furnas, G. W., Landauer, T. K., and Harshman, R. 1990. Indexing by latent semantic analysis. Journal of the American Society for Information Science, 3(41).
9. Kim, S.-M. and Hovy, E. 2006. Automatic identification of pro and con reasons in online reviews. Proceedings of the COLING/ACL Main Conference Poster Sessions, pages 483.
10. Kim, J., Li, J.-J. and Lee, J.-H. 2006. Evaluating Multilanguage-Comparability of Subjectivity Analysis Systems. Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pages 595 Uppsala, Sweden, 11-16 July 2010.
11. P. Koehn and H. Hoang and A. Birch and C. Callison-Burch and M. Federico and N. Bertoldi and B. Cowan and W. Shen and C. Moran and R. Zens and C. Dyer and O. Bojar and A. Constantin and E. Herbst 2007. Moses: Open source toolkit for statistical machine translation. Proceedings of the Annual Meeting of the Association for Computational Linguistics, demonstration session, pages 177–180. Columbus, Oh, USA.
12. Mihalcea, R., Banea, C., and Wiebe, J. 2009. Learning multilingual subjective language via cross-lingual projections. Proceedings of the Conference of the Annual Meeting of the Association for Computational Linguistics 2007, pp.976-983, Prague, Czech Republic.
13. F. J. Och 2003. Minimum error rate training in statistical machine translation. Proceedings of the 41st Annual Meeting on Association for Computational Linguistics, pages 160–167. Sapporo, Japan.
14. Platt, J. C. 1999. Fast training of support vector machines using sequential minimal optimization. Advances in kernel methods, isbn 0-262-19416-3, pages 185–208.
15. K. Papineni and S. Roukos and T. Ward and W. J. Zhu 2001. BLEU: a method for automatic evaluation of machine translation. Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, pages 311–318. Philadelphia, Pennsylvania.
16. Pang, B. and Lee, L. 2008. Opinion Mining and Sentiment Analysis. Found. Trends Inf. Retr., vol. 1, nr. 1–2, 2008.
17. J. Savoy, and L. Dolamic. 2009. How effective is Google's translation service in search?. Communications of the ACM, 52(10):139–143.
18. Steinberger, J. and Lenkova, P. and Ebrahim, M. and Ehrman, M. and Hurriyetoglu, A. and Kabadjov, M. and Steinberger, R. and Tanev, H. and Zavarella, V. and Vazquez, S. 2011. Creating Sentiment Dictionaries via Triangulation. Proceedings of the 2nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis, Portland, Oregon.
19. Steinberger, J. and Lenkova, P. and Kabadjov, M. and Steinberger, R. and van der Goot, E. 2011. Multilingual Entity-Centered Sentiment Analysis Evaluated by Parallel Corpora. Proceedings of the Conference on Recent Advancements in Natural Language Processing (RANLP), Hissar, Bulgaria.
20. Turchi, M. and Atkinson, M. and Wilcox, A. and Crawley, B. and Bucci, S. and Steinberger, R. and Van der Goot, E. 2012. ONTS:"Optima" News Translation System. Proceedings of EACL 2012, pages 25–31.
21. Wan, X. and Li, H. and Xiao, J. 2010. Cross-language document summarization based on machine translation quality prediction. Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pages 917–926.
22. Wilson, T., Wiebe, J., and Hoffmann, P. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. Proceedings of HLT-EMNLP 2005, pp.347-354, Vancouver, Canada.

Towards an Abstractive Opinion Summarisation of Multiple Reviews in the Tourism Domain

Cyril Labbé and François Portet

Laboratoire d'Informatique de Grenoble, UJF/Grenoble-INP/CNRS 5217, 38041 Grenoble, France
first.last@imag.fr

Abstract. Since the arrival of Web 2.0, there is an increasing amount of on-line Reviews and Ratings about diverse products or services. The reviews contain general comments as well as highly personal elements or opinions about the customers' experience with the product. Other customers or companies are facing the problem of extracting the relevant information from this mass of reviews. In this paper, we present a comparative study of three different summarisation techniques for reviews analysis. From this study, we propose a general architecture which relies on a customisable abstractive summarisation approach making use of domain knowledge and temporal analysis. The paper ends by identifying research directions for improving the efficiency of review summarisation methods.

Keywords: Review summarisation, Opinion mining, Natural language generation.

1 Introduction

Since the arrival of Web 2.0, costumers of any kind of products or services produce a large amount of on-line Reviews, almost only present as text, and Ratings as ordinal variable. While these reviews have largely contributed to the success of the e-commerce, the problem for a costumer to construct her/his own opinion and to make an informed decision is to make sense of this mass of reviews that contains not only general comments about product features but also highly idiosyncratic information such as opinions or sentiments. Reviews are not only useful for potential costumers but also represent precious information for companies about their own products. A major challenge for society is to make possible an automatic analysis of sets of reviews in order to produce a coherent summary that can be quickly and easily assimilated by humans.

In this paper, we study the problem of review summarisation in the accommodation domain. Automatic summarisation is the process of drawing out the most relevant information from a source to produce a condensed version sometimes biased towards particular users and tasks. Summarisation approaches are generally categorised as: *extractive* when content reduction is addressed by selection or *abstractive* when compression is done by generalisation of what is relevant in the source [1]. While summarisation of technical structured contents is a well implanted technique in industry, summarisation of reviews is a much more recent trend. In this context, the task must face poorly structured contents from a large number of authors (e.g., age, sex, literacy level, etc.) full of subjective matters expressed via opinion, metaphor, or cultural references. The excerpts (table 1) from an existing database illustrate the variety of reviews for a same hotel.

The hotel was well serviced by friendly staff. Great bathroom with a flat floor shower...... no bath! Mini kitchenette was really handy, great not to have to use the vanity basin as a kitchen sink! While there are no retail shops close by, there is a convenience store next door.Two food courts that have huge variety and restaurants in the Casino to suit any taste & wallet.... are less than 10 min walk away.

The staff from Front Desk to cleaners could not be faulted... friendly and helpful, making us feel like welcome guests.

Booked by work for it’s location this was a rather expensive XXX YYY stay. Wifi was provided by an external provider with very expensive rates. This is not great for people on business. The room was nothing special with a standard shower and mediocre bed. Clean but pretty bog standard. Nothing to rave about and equally nothing terrible to report.

Fig. 1. Example of reviews containing poorly structured content, subjective sentiments and opinions, metaphor, or cultural references.

In this context, sentiment analysis must play a major role when summarising reviews. Sentiment analysis task can be decomposed in several steps. As a first step, analysis of small texts (phrases, tweets, SMS messages) gives the trend of the conveyed sentiment (commonly refereed to as polarity) generally classified as: positive, negative or neutral. Further steps are needed to summarise the global sentiments. The main difficulty is to give a fair and non-biased picture of the global feeling emerging from individual sentiments. This global picture can consist in a set of numbers (tables, charts, graph...) or in a short text that gives the global sentiment in a concise way.

In this study, we propose to compare three approaches to summarisation in order to draw out their current limitations and advantages for this task. This comparison is described in Section 3. Based on this comparison, we propose in Section 4 a new architecture for review summarisation which relies on an abstractive approach making use of domain knowledge and temporal analysis. We conclude the paper with an description of research directions for improving the efficiency of review summarisation methods.

2 Related Research

Summarising opinions reviews into texts can be done in several ways. The most straightforward being the use of a general summariser. Other approaches proposed to produce a tailored “voice summary” of a set of the most extreme restaurants reviews [2]. In the tailored-summariser ReSum [3] the target are reviews on products sold on-line. ReSum outputs two summaries, one for the positive reviews, one for the negative reviews. These summaries are composed of sentences extracted from the positive (or negative) reviews according to a strategy involving redundancy elimination and domain-feature depend criteria such as technical level or Time of Ownership. Here and in the following, features refer to domain characteristics. For instance, in the accommodation domain, quality of beds or cleanness of the room are domain features (or aspects). While these approaches provide interesting summaries they do not consider opinions in a systematic way, hence the need for a sentiment analysis module in the summarisation framework.

Sentiment summarisation involves several steps (for a detailed review the reader is referred to [4]). The first step aims at determining the sentiment express by each individual reviews. Representative examples of this step are [5] and [6]. [5] proposes a method for determining opinion polarity using WordNet, SentiWordNet and the General Inquirer (to detect polarity shifter). In [6], The probability $P(+/rv)$ (resp $P(-/rv)$) of a movie review rv of being a review of positive (resp. negative) polarity is estimated through the use of Naive Bayes and Markov Model techniques. Each individual review is then scored and this score is used to retrieve the most extreme reviews. However, the method does not capture the global sentiment emerging from the reviews.

The global sentiment of a set of reviews can be abridged as numbers or charts. For example, [7] summarises hotel reviews through automatic features extractions and polarity measure. For each review, if a feature is identified, its polarity is computed. The global sentiment for each feature is then computed. In [8], reviews are summarised in a similar manner but using a domain ontology for features identification. An important advantages of this approach is that it proposes to highlight positive (reps. negative) comments within negative (resp. positive) reviews arguing that opinions about features are more interesting when extracted from a review containing contrasted opinions.

The next section gives a more detailed focus on "pro" and "cons" associated to three methods for the summarisation of the global sentiment emerging from hotel reviews.

3 Comparative studies of three approaches

Three different approaches used to summarise the overall opinion emerging from a set of hotel reviews are presented. The reviews were all collected from the Tripadvisor website. The first experiment concerns the use of a general summariser. The second one shows results obtained when sentences extraction is guided by domain features. The third one consists in the Reviews and Ratings (RnR) system described in [8].

3.1 Open Text Summarizer

Open Text Summarizer (OTS) is an open source tool for summarising texts of any domain [9]. Its content selection is based on the TF-IDF measure with some re-weighting based on the structure of the document (e.g., title and paragraph). The experiment with OTS consisted in feeding it with a whole set of reviews about the same hotel and checking the output. Figure 2 shows an output when OTS was applied with a 1% compression ratio. It can be noticed that no relevant information about the hotel appears before the fourth sentences. As with any extractive summariser, some referring expressions are impossible to understand (e.g.,"**This** appeared from the unlocked..."). Moreover, there is no way for the summariser to filter out irrelevant information for the decision making task such as with information about booking experience (e.g., "booking was done at very last minute...", "I did a lot of research..."). This is due to the high frequency of personal booking experiences that biased the system towards this kind of irrelevant information. It appears from this short example, that purely frequency-based content selection without the involvement of some domain and/or task knowledge is unpromising.

```
Hotel booking was done at very last minute by the friendly staff at the
International Airport. I did a lot of research in advance - most of it on
Tripadvisor - and it was ranked very highly. This appeared from the unlocked
office behind reception - I was told this was more secure - I wondered but all was
ok. Location is what this hotel has going for it - you're on holidays, you want to
be in the centre of things, near good restaurants [...]
```

Fig. 2. Beginning of the output of OTS at 1% compression rate (2500 to 17 sentences).

3.2 Features-based Selection Extraction

In this approach, the main idea is to extract relevant information related to a particular word. In [10], an approach to better understand the particular meaning associated to a word in the mind of a particular author was proposed. We proposed to use this technique to capture the global opinion given by a set of users on a particular domain feature.

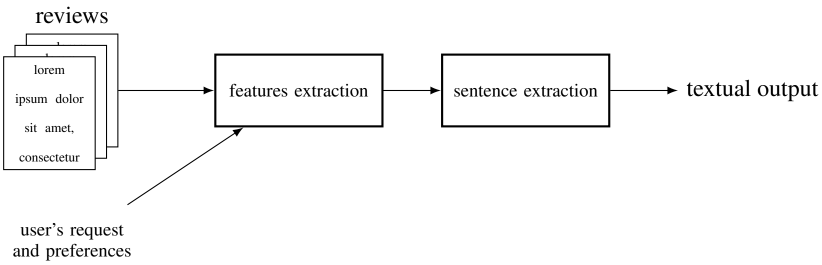


Fig. 3. Diagram of the extractive system

Figure 3 shows the outline of the proposed method. Let denote by f a word type representing a particular feature under study (BED for example). The set of sentences containing f is a sub-corpus denoted by U_f and is called the *lexical universe of* f. For each word type i of the global corpus (C) under consideration two frequencies can be observed:

- $p_{U_f}(i)$ is the observed frequency of type i in U_f the lexical universe of f.
- $p_C(i)$ is the frequency of type i in the whole corpus.

Using the hypergeometric law, an expected value $E_{U_f}(i)$ for $p_{U_f}(i)$ can be computed. Given a confidence level (5% or 1%) it is then possible to tell if the observed value $p_{U_f}(i)$ is too *far* from $E_{U_f}(i)$, either because $p_{U_f}(i) << E_{U_f}(i)$ or because $p_{U_f}(i) >> E_{U_f}(i)$. So each word type i of the whole corpus C can be classified, with regards to f as being:

- *neutral*, this is a set of words that do not have a special interaction with f. There frequency in the lexical universe of f is acceptable with regards to their frequency in the whole corpus;
- *attracted* (if $p_{U_f}(i) >> E_{U_f}(i)$), this is the set of words that are over-represented in the lexical universe of f. They can be seen as being attracted by f and it can be inferred that they are characterizing the global opinion on f;
- *repulsed* (if $p_{U_f}(i) << E_{U_f}(i)$), this is the set of words that are under-represented in the lexical universe of f. They can be seen as being repulsed by f and it can be inferred that they are not reflecting the global opinion on f;

Given a feature f it is then possible to build two sets. U_f^+ the set of words that mostly *characterize* f and U_f^- the set of words that are mostly *repulsed* by f. These sets are used to score each sentences of the whole corpus C so to select the set of sentences that characterize the best the opinion associated to a particular feature.

Figure 4 shows the most relevant sentences for the feature BED. It can be noticed that the most relevant and condense sentences are the best rated. However, there is a high redundancy in this list and contrasted reviews are not fetched by the method.

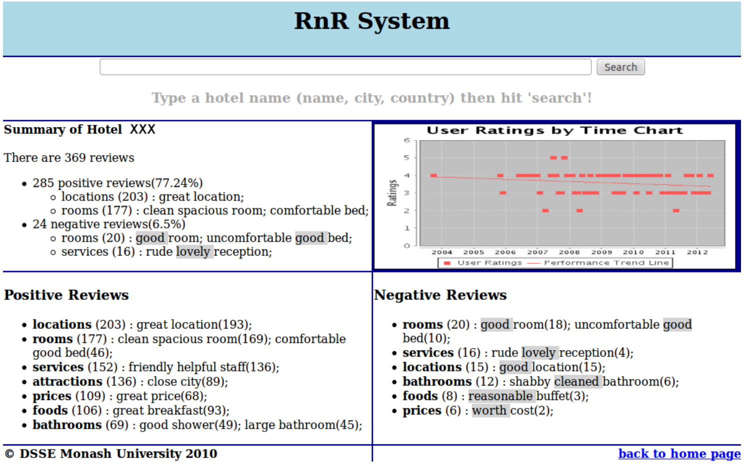
0.647	A comfortable double bed, couch and coffee table, plus a small desk with two chairs.
0.615	The room was spacious with a queen sized bed and a sofa bed.
0.412	The hotel rooms were a good size with a double bed and a fould out sofa bed.
0.378	Queen sized bed (with small side shelfs), little couch and coffee table for persons, a basic table with chairs, a flat screen tv, and a dresser with a couple of drawers.
0.370	The rooms were quite large - we had a queen room which consisted of a queen bed, small lounge, small table and chairs and kitchenette.
0.364	The room was quite large with a couch, desk and amp ; coffee table as well as the queen size bed.
	. . .

Fig. 4. Example of extracted sentences for the feature BED.

3.3 RnR system

In [8], an RnR system [1] for extracting rationale from on-line reviews/ratings is presented. The system captures and summarises the key rationale for positive and negative opinions expressed in a corpus of reviews and highlights the negative features among positive reviews and vice versa. One of the main contribution of the work is the techniques that have been designed to leverage support metric in conjunction with a domain ontology. This results in improved computational overheads associated with sentiment identification. In term of presentation, the system outputs the summary for each hotel in a four-quarter screen presented in Figure 5. The top left quarter shows the general/summarised overview of the hotel, top right column contains the time based performance chart, and the two bottom sections give details of each positive (left hand side) and negative (right hand side) groups of reviews.

[1] The RnR system is accessible at http://rnrsystem.com/RnRSystem



RnR System

Search

Type a hotel name (name, city, country) then hit 'search'!

Summary of Hotel XXX

There are 369 reviews

- 285 positive reviews(77.24%)
  - locations (203) : great location;
  - rooms (177) : clean spacious room; comfortable bed;
- 24 negative reviews(6.5%)
  - rooms (20) : good room; uncomfortable good bed;
  - services (16) : rude lovely reception;

Positive Reviews

- **locations** (203) : great location(193);
- **rooms** (177) : clean spacious room(169); comfortable good bed(46);
- **services** (152) : friendly helpful staff(136);
- **attractions** (136) : close city(89);
- **prices** (109) : great price(68);
- **foods** (106) : great breakfast(93);
- **bathrooms** (69) : good shower(49); large bathroom(45);

Negative Reviews

- **rooms** (20) : good room(18); uncomfortable good bed(10);
- **services** (16) : rude lovely reception(4);
- **locations** (15) : good location(15);
- **bathrooms** (12) : shabby cleaned bathroom(6);
- **foods** (8) : reasonable buffet(3);
- **prices** (6) : worth cost(2);

© DSSE Monash University 2010

back to home page

Fig. 5. RnR output.

Though the RnR output provides the useful global picture of the reviews, it is lacking a fundamental dimension which is the temporal dimension. The rating chart does indeed give trends but is of little interest when the tend is flat as it the case in Figure 5. So there is no way for the costumer to know the latest positive and negative features of the hostel nor to know what are the positive and negative constants of it. Furthermore, tabular and keyword presentation might not be the best way of presenting a summarisation of the reviews as every pieces of information is presented in an out-of-context way. A more elegant approach to present such information both with respect to the temporal and contextual perspectives is to use Natural Language Generation (NLG).

4 Towards an abstractive summarisation system

NLG systems has been used for decade to present numerical and linguistic information in a condense and efficient way. Recently, NLG has been applied to summarise large volumes of heterogeneous temporal data to short texts in the medical domain [11]. This system was experimented at the hospital and has shown that a textual-only output can led to better decision from the medical staff than a classical graphical-only presentation. Among the properties, emphasised by the authors [12], that textual summarisation offers compared with the graphical presentation are : the capacity to present data in the same sentence at multiple time resolution or period (e.g., “the hotel had always been praised for its good beds”, “in summer, the hotel is found to be badly ventilated”), the natural ability to handle vagueness and uncertainty (e.g., “the hotel seems to be close to public transport”), the capacity to insert genuine citations (e.g., “the hotel could not even offer us a hand towel!”), the possibility to aggregate features (e.g., “close station(90); free tram(44); close train(33);” → “close public transport and free tram”) and the capacity to contrast features (e.g., “even the negatives reviews reports that the bathroom is generally clean and large”).

To address the above limitations and progress beyond the state of the art in this domain, we plan to build an approach based on the work of Rahayu *et al.* [8] and Portet *at al.* [11]. This approach combines a sentiment analyses and domain-specific text processing approaches to represent the data in a high level representation (e.g., in the form of an ontology) with a natural language generation system to generate a textual user-tailored review of an hostel. The intended system is depicted Figure 6. User requests a summary of a specific hostel. In some cases, she can also specify which features are the most important for her so that features belonging to her preferences are given more weight. The system then fetches all the opinions about this hotel (e.g., trip advisor) and extract the features describing each reviews. Once the features extractions is performed, a sentiment analysis layer extracts polarity affecting each phrases of interest. These phrases are then abstracted into facts in a database backed by an ontology which represent the hotel and customer's concepts. Using the ratings, a time series segmentation [13] is performed to identify the main periods of the hotel (decrease, increase, stable). Another segmentation is performed at the feature level to detect specific evolutions of the hotel's services. Once the opinions have been analysed all the data is summarised through an NLG approach.

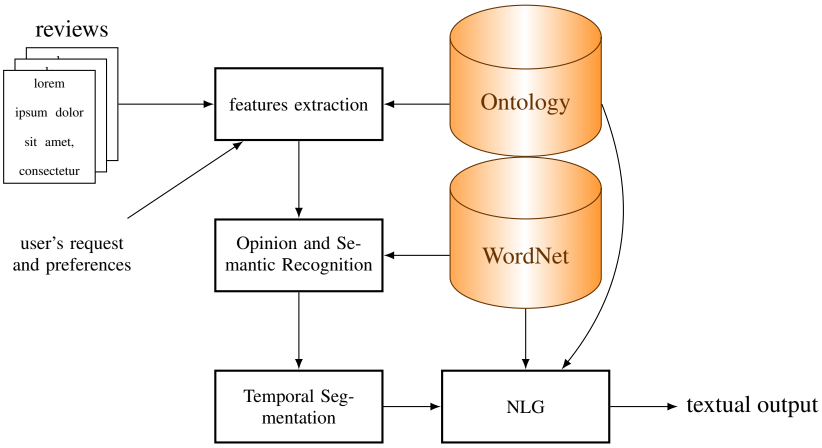


Fig. 6. Diagram of the abstractive system

5 Conclusion

Although human summaries are typically abstracts, most existing systems produce extracts, due to several studies reporting better results of the latter [1]. This is due to the complexity the process that involves concepts extraction, reasoning at the semantic level and natural language generation. This makes it a time consuming task. However, review summarisation is a very different application than documents considered in classical summarisation. The high number of authors, style, subjectivity and the temporal dimension calls for the reconsideration of the abstractive approaches to perform a deep analysis to better condense the information present in the reviews. Our approach by considering these aspect while aiming for a modular architecture, is a step towards addressing this challenge.

Another important challenge is the evaluation of such technology. This is delicate given that no gold standard summary exists in this domain for automatic scoring (such as with BLEU or ROUGE) and because users will often disagree on what constitutes the best content and quality for the summary. A more relevant measure would be to perform some task-based experiments to assess the effectiveness of the summariser in searching for an hotel. We plan to investigate the techniques used in different domains to propose a formal evaluation strategy which would make it possible to assess the progress of the method.

References

1. Mani, I., Maybury, M., eds.: Advances in Automatic Text Summarization. MIT Press (1999)
2. Mahajan, M., Nguyen, P., Zweig, G.: Summarization of multiple user reviews in the restaurant domain. Technical Report MSR-TR-2007-126, Microsoft Research (2007)
3. Kokkoras, F., Lampridou, E., Ntonas, K., Vlahavas, I.: Summarization of multiple, metadata rich, product reviews. In: Workshop on Mining Social Data (MSoDa), 18th European Conference on Artificial Intelligence (ECAI '08), Patras, Greece (2008)
4. Tang, H., Tan, S., Cheng, X.: A survey on sentiment detection of reviews. Expert Syst. Appl. **36**(7) (September 2009) 10760–10773
5. Martín-Wanton, T., Pons-Porrata, A., Montoyo-Guijarro, A., Balahur, A.: Opinion polarity detection - using word sense disambiguation to determine the polarity of opinions. In Filipe, J., Fred, A.L.N., Sharp, B., eds.: ICAART (1). (2010) 483–486
6. Salvetti, F., Lewis, S., Reichenbach, C.: Automatic opinion polarity classification of movie reviews. Colorado Research in Linguistics **17**(1) (2004)
7. Popescu, A.M., Etzioni, O.: Extracting product features and opinions from reviews. In: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing. HLT '05, Stroudsburg, PA, USA, Association for Computational Linguistics (2005) 339–346
8. Rahayu, D., Krishnaswamy, S., Labbé, C., Alahakoon, O.: Web services for analysing and summarising online opinions and reviews. In: ServiceWave. (2010)
9. Rotem, N.: Open text summarizer (ots) (2003) Retrieved June, 2012, http://libots.sourceforge.net.
10. Labbé, C., Labbé, D.: How to measure the meanings of words? Amour in Corneille's work. Language Resources and Evaluation **39**(4) (2005) 335–351
11. Portet, F., Reiter, E., Gatt, A., Hunter, J., Sripada, S., Freer, Y., Sykes, C.: Automatic generation of textual summaries from neonatal intensive care data. Artificial Intelligence **173**(7–8) (2009) 789–816
12. Gatt, A., Portet, F., Reiter, E., Hunter, J., Mahamood, S., Moncur, W., Sripada, S.: From data to text in the neonatal intensive care unit: Using NLG technology for decision support and information management. AI Communications **22**(3) (2009) 153–186
13. Charbonnier, S., Portet, F.: A self-tuning adaptive trend extraction method for process monitoring and diagnosis. Journal of Process Control **22** (2012) 1127–1138