Peggy Cellier
Thierry Charnois
Andreas Hotho
Stan Matwin
Marie-Francine Moens
Yannick Toussaint (Eds.)

Interactions between Data Mining and Natural Language Processing

3rd International Workshop, DMNLP 2016
Riva del Garda, Italy, September 23, 2016
Proceedings

Volume Editors

Peggy Cellier
INSA Rennes, IRISA
Campus Beaulieu, 35042 Rennes cedex, France
E-mail: peggy.cellier@irisa.fr

Thierry Charnois
Université Paris 13 Sorbonne Paris Cité, LIPN CNRS
Av. J.B. Clément, 93430 Villetaneuse, France
E-mail: Thierry.Charnois@lipn.univ-paris13.fr

Andreas Hotho
University of Würzburg
Am Hubland, 97074 Würzburg, Germany
E-mail: hotho@informatik.uni-wuerzburg.de

Stan Matwin
Faculty of Computer Science, Dalhousie University
6050 University Ave., PO BOX 15000, Halifax, NS B3H 4R2, Canada
E-mail: stan@cs.dal.ca

Marie-Francine Moens
Department of Computer Science, KU Leuven
Celestijnenlaan 200A, B-3001 Heverlee, Belgium
E-mail: sien.moens@cs.kuleuven.be

Yannick Toussaint
INRIA Nancy Grand-Est, LORIA
615 Rue du Jardin Botanique, 54600 Villers-lès-Nancy, France
E-mail: Yannick.Toussaint@loria.fr

Preface

Recently, a new field has emerged taking benefit of both domains: Data Mining (DM) and Natural Language Processing (NLP). Indeed, statistical and machine learning methods hold a predominant position in NLP research[1], advanced methods such as recurrent neural networks, Bayesian networks and kernel based methods are extensively researched, and "may have been too successful (...) as there is no longer much room for anything else"[2]. They have proved their effectiveness for some tasks but one major drawback is that they do not provide human readable models. By contrast, symbolic machine learning methods are known to provide more human-readable model that could be an end in itself (e.g., for stylistics) or improve, by combination, further methods including numerical ones. Research in Data Mining has progressed significantly in the last decades, through the development of advanced algorithms and techniques to extract knowledge from data in different forms. In particular, for two decades Pattern Mining has been one of the most active field in Knowledge Discovery.

This volume contains the papers presented at the ECML/PKDD 2016 workshop: DMNLP'16, held on September 23, 2016 in Riva del Garda. DMNLP'16 (Workshop on Interactions between Data Mining and Natural Language Processing) is the third edition of a workshop dedicated to Data Mining and Natural Language Processing cross-fertilization, *i.e* a workshop where NLP brings new challenges to DM, and where DM gives future prospects to NLP. It is well-known that texts provide a very challenging context to both NLP and DM with a huge volume of low-structured, complex, domain-dependent and task-dependent data. The objective of DMNLP is thus to provide a forum to discuss how Data Mining can be interesting for NLP tasks, providing symbolic knowledge, but also how NLP can enhance data mining approaches by providing richer and/or more complex information to mine and by integrating linguistic knowledge directly in the mining process. Out of 12 submitted papers, 6 were accepted.

The high quality of the program of the workshop was ensured by the much-appreciate work of the authors and the Program Committee members. Finally, we wish to thank the local organization team of ECML/PKDD 2016. and the ECML/PKDD 2016 workshop chairs Matthijs van Leeuwen, Fabrizio Costa, and Albrecht Zimmermann.

September 2016

Peggy Cellier, Thierry Charnois
Andreas Hotho, Stan Matwin
Marie-Francine Moens, Yannick Toussaint

[1] D. Hall, D. Jurafsky, and C. M. Manning. Studying the History of Ideas Using Topic Models. In Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing, pp. 363–371, 2008

[2] K. Church. A Pendulum Swung Too Far. Linguistic Issues in Language Technology, Vol. 6, CSLI publications, 2011.

Organization

Program Chairs

Peggy Cellier	INSA Rennes, IRISA, France
Thierry Charnois	Université Paris 13, Sorbonne Paris cité, LIPN, France
Andreas Hotho	University of Kassel, Germany
Stan Matwin	Dalhousie University, Canada
Marie-Francine Moens	Katholieke Universiteit Leuven, Belgium
Yannick Toussaint	INRIA Nancy Grand-Est, LORIA, France

Program Commitee

Martin Atzmueller	University of Kassel, Germany
Delphine Battistelli	MoDyCo-Université Paris Ouest, France
Yves Bestgen	F.R.S-FNRS, Université catholique de Louvain, Belgium
Bruno Crémilleux	Universit de Caen, France
Béatrice Daille	Laboratoire d'Informatique de Nantes Atlantique, France
Luigi Di Caro	University of Torino, Italy
Jiří Kléma	Czech Technical University, Prague, Czech Republic
Yves Lepage	Waseda University, Japan
Amedeo Napoli	LORIA Nancy, France
Claire Nédellec	Institut National de Recherche Agronomique, France
Maria Teresa Pazienza	University of Roma "Tor Vergata", Italy
Pascal Poncelet	LIRMM Montpellier, France
Solen Quiniou	Laboratoire d'Informatique de Nantes Atlantique, France
Mathieu Roche	Cirad, TETIS, Montpellier, France
Christin Seifert	Universitat Passau, Germany
Arnaud Soulet	Université François Rabelais Tours, France
Koichi Takeuchi	Okayama University, Japan
Isabelle Tellier	Lattice, Paris, France

Table of Contents

Using Machine Learning Methods and Linguistic Features in Single-Document Extractive Summarization

Alexander Dlikman and Mark Last

Department of Information Systems Engineering
Ben-Gurion University of the Negev
Beer-Sheva, Israel
dlikman@post.bgu.ac.il, mlast@bgu.ac.il

Abstract. Extractive summarization of text documents usually consists of ranking the document sentences and extracting the top-ranked sentences subject to the summary length constraints. In this paper, we explore the contribution of various supervised learning algorithms to the sentence ranking task. For this purpose, we introduce a novel sentence ranking methodology based on the similarity score between a candidate sentence and benchmark summaries. Our experiments are performed on three benchmark summarization corpora: DUC-2002, DUC-2007 and MultiLing-2013. The popular linear regression model achieved the best results in all evaluated datasets. Additionally, the linear regression model, which included POS (Part-of-Speech)-based features, outperformed the one with statistical features only.

Keywords: text summarization, part-of-speech tagging, supervised learning, regression, sentence ranking

1 Introduction

In this study, we seek to improve the performance of extractive summarization algorithms by using multiple statistical and linguistic sentence features combined with advanced machine learning techniques. We apply the following four supervised learning algorithms to the extractive summarization task: Classification and Regression Trees (CART) [3], Cubist [9], linear regression, and a genetic algorithm. The algorithms are trained on benchmark corpora of summarized documents and compared to state-of-the-art extractive summarization tools using the same feature sets. The proposed supervised methodology for sentence extraction is based on a continuous similarity score between candidate sentences and human-generated gold standard summaries. For this purpose, a novel, Penalized Precision metric is introduced.

In: P. Cellier, T. Charnois, A. Hotho, S. Matwin, M.-F. Moens, Y. Toussaint (Eds.): Proceedings of DMNLP, Workshop at ECML/PKDD, Riva del Garda, Italy, 2016.

2 Related Work

2.1 Extractive Text Summarization

Extractive summarization techniques identify the most important sentences in the input text(s) and combine them to create a summary of a pre-defined length. Various sentence scoring metrics, or features, have been proposed in literature. Gupta and Lehal [7] in their survey of text summarization techniques list the following groups of features: keyword-based, title-based, location-based, length-based, proper noun and upper-case word-based, font-based, specific phrase-based, and features based on the sentence similarity to other sentences in text. The MUSE summarization algorithm [15, 14] is a representative example of an extractive summarizer, built upon 31 statistical sentence metrics. These metrics are divided into *structure-based*, *vector-based* and *graph-based* groups. The MUSE summarizer uses a supervised approach with Genetic Algorithm to find the best feature weights from a given corpus of summarized documents.

Several extractive summarization approaches make use of linguistic sentence scoring metrics for text representation and calculation of the final sentence score. The most typical approach is the use of proper nouns or upper case words [7, 11, 12]. Fattah and Ren [5] use the count of numerical data and proper noun occurrences in a sentence. Al-Hashemi [2] employs human-generated rules based on POS (Part-of-Speech) sequences in an extractive summarization system. Mihalcea and Tarau [18] present a graph-based model for keyword extraction which makes use of POS tags. In this approach, a graph represents the text and interconnects words or other text entities. The authors propose several options including all words, only nouns, only nouns and verbs or only nouns and adjectives. One of conclusions of Mihalcea and Tarau's study shows that the performance of models without POS information is significantly lower than those that consider POS information.

2.2 Machine Learning Methods for Sentence Extraction

In the regression approach to the sentence ranking task, the score of each candidate sentence s is evaluated as a weighted average of all its features [20]. The feature weights can be found by various machine learning techniques such as a linear regression [5] or a Genetic Algorithm [14]. Ouyang et al. [21] apply a Support Vector Regression (SVR) model to the task of query-based, multi-document extractive summarization. Their SVR framework is based on a set of seven sentence features. Galanis et al. [6] present an Integer Linear Programming (ILP) based approach for extractive, query-based multi-document summarization. The proposed method simultaneously maximizes both the importance of the sentences that are included in a summary as well as their diversity. In order to find a sentence's importance score sentence, the authors use SVR model based on five various predictors (sentence features). The "true" importance (outcome of the regression) is obtained as a ROUGE score between candidate sentences and human-generated summaries.

Compared to other regression-based summarization methods that use seven predictive features in [21] and five in [6], we employ a much larger set of sentence scoring

metrics (30 statistical features from [14] and 17 novel linguistic features) and perform feature selection to preserve the most important features in the model. In addition, both [21] and [6] utilize a sentence-to-summary similarity score, which prefers the longest sentences in the extraction stage. The sentence-to-summary similarity score proposed in our study (Penalized Precision) handles this limitation and penalizes both "too short" and "too long" sentences.

3 Methodology

3.1 Linguistic features

In this section, we introduce 17 POS-based sentence features, which are listed in **Table 1**. Some of them are completely novel while others are derived from our interpretation of certain metrics used by Litvak and Last [14] in the MUSE summarizer. All proposed POS features take into account only nouns, verbs, adjectives and adverbs due to the semantic importance of these parts of speech [13]. These features can be divided into *POS ratio-based* (defined as a ratio between the number of the above parts-of-speech in a sentence and the sentence length); *POS filtering* (employing the original MUSE features after keeping the above POSs and discarding the rest of the words); and *POS patterns* (these features take into account part-of-speech n-grams, which are frequent in human-generated summaries and, at the same time, relatively rare in the original texts).

While the first two methods do not need further explanation, the POS pattern metrics are defined as follows. We assume that the presence of a specific POS pattern in a candidate sentence may indicate sentence relevance in the summary [2]. Our method requires a preprocessing stage where the relevance of the candidate POS patterns is calculated. We define POS pattern relevance as a ratio between normalized pattern frequency in human-generated summaries and normalized pattern frequency in the corpus. The measure is greater than one when the POS n-gram is relatively more frequent in summaries than in the original texts. In the last stage, we sum up all POS n-gram relevance measures, which are greater than one, and normalize this value by the total amount of n-grams in a sentence. In the current work, we calculated the above metrics separately for 2-, 3- and 4- POS grams.

3.2 Sentences Ranking

Our methodology for the sentence ranking task includes the following steps: data preparation, calculation of sentence similarity to benchmark summaries, data scaling, training, and evaluation.

Data Preparation: In the data preparation stage, we generate a sentence-feature matrix for the training corpus. Each row of the matrix refers to a sentence i; each column refers to a feature; and entry of the matrix (m_{ij}) indicates the score of feature j for sentence i.

Each sentence is associated with $sentence\_ID$ and $document\_ID$. The feature set includes the original, language-independent MUSE features as well as our novel linguistic features.

Category	Feature	Description
POS Ratio-Based	POS_NN_RATIO	Ratio of *nouns* to all words in the sentence
	POS_VB_RATIO	Ratio of *verbs* to all words in the sentence
	POS_JJ_RATIO	Ratio of *adjectives* to all words in the sentence
	POS_RB_RATIO	Ratio of *adverbs* to all words in the sentence
POS Filtering	POS_V_TITLE_O	Overlap similarity to the document title
	POS_V_TITLE_J	Jaccard similarity to the document title
	POS_V_TITLE_C	Cosine similarity to the document title
	POS_V_TF	Average term frequency for all POS words
	POS_V_COV	Coverage of POS keywords
	POS_V_TFISF	Sum of term frequencies times inverse sentence frequencies
	POS_V_KEY	Sum of POS keyword frequencies
	POS_V_D_COV_O	Overlap similarity to the document complement
	POS_V_D_COV_J	Jaccard similarity to the document complement
	POS_V_D_COV_C	Cosine similarity to the document complement
POS Patterns	POS_N2	POS 2-gram relevance measure
	POS_N3	POS 3-gram relevance measure
	POS_N4	POS 4-gram relevance measure

Table 1. Part-of-Speech features

Sentence to Summary Similarity Score: The most complex stage is determining the similarity between each sentence and a gold standard summary of the corresponding document. Similarity measures such as ROUGE and other recall-based measures, which normalize joint terms between sentence and benchmark summaries by a summary length, prefer longer sentences by assigning them a higher score. On the other hand, precision-based measures, which normalize joint terms by sentence length, prefer shorter sentences.

To address those issues, we have modified the *BLEU (Bilingual Evaluation Understudy)* measure, which originally was used for evaluating the quality of machine translation [22]. Our implementation of the BLEU score (Eq. 1) is precision penalized when a sentence is "too short".

$$PenPr = P * pen$$

$$pen = \begin{cases} 1 & if\ length(s) > min.length \\ e^{1-\frac{min.length}{length(s)}} & if\ length(s) \leq min.length \end{cases} \quad (1)$$

P stands for the sentence precision, which naturally penalizes "too long" sentences as well, and the $min.length$ parameter represents the minimum sentence length in a gold standard summary. When several benchmark summaries exist per each document, we calculate the $PenPr$ value for each benchmark summary separately and then provide the average similarity of a sentence to benchmark summaries, exactly as in the ROUGE method.

Data Scaling: The max-min rescaling method is used to normalize the feature values to the [0, 1] range based on their minimum and maximum values in the training corpus. In contrast, to normalize the values of sentence similarity to the gold standard, we calculate the minimum and maximum similarity values separately for each document. This approach allows to deal with the fact that gold standard summaries in the corpus can be both extractive and abstractive (for extractive summaries, the similarity values tend to be higher than for the abstractive ones).

Training: By using the columns in the sentence-feature matrix as regression predictors and sentence similarity to the gold standard as a continuous target variable, any regression algorithm can be trained. The resulting regression model will include the values of the feature weights.

Evaluation. To evaluate the performance of the induced model on a hold-out set, we first compute the predicted value of each sentence similarity score ($\hat{y}$). After this, n top ranking sentences (based on $\hat{y}$) are extracted to a peer summary, subject to a summary length constraint. The resulting peer summaries can be evaluated using various ROUGE measures and available gold standard summaries.

4 Evaluation Experiments

4.1 Datasets and Software Tools

For training and testing, we used three different English corpora containing summarized documents. *DUC-2002* [4], which was prepared for the summarization competition task at the Document Understanding Conference, is a gold-standard dataset that contains 531 news articles from the Wall Street Journal (1987-1992), and the Financial Times (1991-1994). Each textual document contains at least 10 sentences and appears with two to three human-generated ("gold standard") abstractive summaries of around 100 words.

An additional evaluated corpus is *DUC-2007* [4]. The main task of DUC-2007 was, given a topic and a set of 25 relevant documents, to synthesize a fluent, well-organized

250-word summary of the documents that would answer the question in the topic statement, i.e., perform a multi-document query-based summarization. Each topic is accompanied with up to four human-generated abstractive summaries of around 250 words. In order to allow single-document training, all documents on a particular topic were merged into one text.

We have also used an English corpus from the MultiLing 2013 single-document summarization task [19]. The dataset includes 30 Wikipedia articles with one gold standard (human-generated) summary of around 270 words per article. Due to relatively small amount of documents, MultiLing-2013 is used only as test data in cross-corpus evaluation experiments.

In our study, we used MUSEEC, an open-source text summarization tool [16]. For the purpose of preprocessing (sentence splitting, tokenization, stop words removal and lemmatization) and part-of-speech tagging, we used the popular *Stanford CoreNLP toolkit* [17], an extensible pipeline that provides core natural language analysis. For sentence ranking, we used several R packages: *GA* Package [23] for Genetic Algorithm, *rpart* [24] for CART algorithm, *cubist* [10] for Cubist algorithm. The *Caret* R package [8] was used for parameter optimization of those algorithms and cross-validation when implementing the experiments described below.

4.2 Evaluation Results

We evaluated four regression approaches to the sentence ranking task: CART [3], LM (linear regression model), GA (Genetic Algorithm) and Cubist [9]. We also compared the results to MUSE [14] as a state-of-the-art supervised method for extractive summarization. Each model was evaluated with four different feature sets: *MUSE* (30 original features used by MUSE); *POS only* (17 POS-based features); *POS Extended* (17 POS-based features + Sentence Position + Sentence Length); and *MUSE & POS* (both MUSE and POS-based features).

DUC-2002 (10-fold cross-validation): Cubist and LM using the most complete feature set (MUSE & POS) were the top ranking approaches. Since the difference between them was not found statistically significant (p-value of 0.205) we preferred the simpler LM approach. In further statistical tests, we compared LM models with different feature sets (the first four rows in **Table 2**). As can be seen from the results, the MUSE & POS feature combination is significantly better than the other feature sets. The subsequent experiments (the last three rows in **Table 2**) compared the LM model with three other models (all using MUSE & POS features). The results are statistically significant and show that LM outperforms all other models. Using the Akaike Information Criterion (AIC) statistics [1] for stepwise feature selection, 4 statistical features (D_COV_J, KEY_DEG, KEY_PR, SVD) and 4 POS-based features (POS_B, POS_RB_RATIO, POS_V_TITLE_C, POS_V_TITLE_O) were discarded as statistically insignificant.

DUC-2007 (10-fold cross-validation): In this dataset, the difference between the MUSE and the MUSE & POS feature sets was not found statistically significant and, thus, the MUSE feature set was preferred due to simplicity. The experiments have shown that the LM model with MUSE features outperforms all other models with the same feature set.

MultiLing-2013 (training on DUC-2002): In the MultiLing-2013 corpus, both Cubist and LM with the MUSE & POS feature set are the top-ranking models, without a statistically significant difference between them. Consequently we prefer the simpler LM approach. The results show that LM with the MUSE & POS feature set outperforms all other models.

Model	Features	ROUGE-1 F	p-value
LM	MUSE & POS	0.464	--
LM	POS Extended	0.460	0.031
LM	MUSE	0.457	0.000
LM	POS only	0.454	0.001
MUSE	MUSE & POS	0.457	0.003
GA	MUSE & POS	0.452	0.000
CART	MUSE & POS	0.444	0.000

Table 2. DUC-2002 results with different feature sets

5 Conclusions

In this work, we have explored the contribution of various machine learning algorithms to sentence ranking and introduced a novel, Penalized Precision metric. The results of our experiments show that in all evaluated textual corpora, the linear model outperforms the more sophisticated CART and Cubist regression models, the heuristic optimization with genetic algorithm, as well as the state-of-the-art summarization approach (MUSE). Additionally, the linear models which included POS features, outperform those with statistical features only. To achieve the best results, we suggest using the Linear Model with statistical and POS-based features. Future work may focus on extending the proposed POS-based features and sentence ranking techniques to other languages and domains.

6 References

1. Akaike, H. 1974. A New Look at the Statistical Model Identification. *IEEE Transactions on Automatic Control* 19 (6): 716-723.
2. Al-Hashemi, R. 2010. Text Summarization Extraction System (TSES) Using Extracted Keywords. *International Arab Journal of e-Technology* 1 (4): 164-168.
3. Breiman, L., Friedman, J., Stone, C., and Olshen, R. 1984. *Classification and regression trees.* CRC press.
4. *Document Understanding Conferences.* http://duc.nist.gov/.

5. Fattah, M. A., and Ren, F. 2009. GA, MR, FFNN, PNN and GMM based models for automatic text summarization. *Computer Speech & Language* 23 (1): 126-144.
6. Galanis, D., Lampouras, G., and Androutsopoulos, I. 2012. Extractive Multi-Document Summarization with Integer Linear Programming and Support Vector Regression. *COLING 2012: Technical Papers.* Mumbai,India. 911-926.
7. Gupta, V., and Lehal, G. 2010. A Survey of Text Summarization Extractive Techniques. *Journal of Emerging Technologies in Web Intelligence* 2 (3): 258-268.
8. Kuhn, M. 2015. *caret: Classification and Regression Training.* http://CRAN.R-project.org/package=caret.
9. Kuhn, M., and Johnson, K. 2013. *Applied predictive modeling.* New York: Springer.
10. Kuhn, M., Weston, S., Keefer, C., and Coulter, N. 2014. *Cubist: Rule- and Instance-Based Regression Modeling.* http://CRAN.R-project.org/package=Cubist.
11. Kupiec, J., Pedersen, J., and Chen, F. 1995. A trainable document summarizer. *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval.* 68-73.
12. Kyoomarsi, F., Khosravi, H., and Eslami, E. 2008. Optimizing Text Summarization Based on Fuzzy Logic. *Seventh IEEE/ACIS International Conference on Computer and Information Science (icis 2008).* 347-352.
13. Lioma, C., and Blanco, R. 2009. Part of Speech Based Term Weighting for Information Retrieval. In *Advances in Information Retrieval*, 412-423. Springer Berlin Heidelberg.
14. Litvak, M., and Last, M. 2013. "Cross-lingual training of summarization systems using annotated corpora in a foreign language." *Information retrieval* 16 (5): 629-656.
15. Litvak, M., Last, M., and Friedman, M. 2010. A new approach to improving multilingual summarization using a genetic algorithm. *48th Annual Meeting of the Association for Computational Linguistics.* 927-936.
16. Litvak, M., Vanetik, N., Last, M., and Churkin, E. 2016. MUSEEC: A Multilingual Text Summarization Tool. to appear in *54th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.
17. Manning, C., Surdeanu, M., and Bauer, J. 2014. The Stanford CoreNLP Natural Language Processing Toolkit. *52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations.* 55-60.
18. Mihalcea, R., and Tarau, P. 2004. TextRank: Bringing order into texts. *Conference on Empirical Methods in Natural Language Processing.* Barcelona, Spain: Association for Computational Linguistics. 404–411 .
19. *MultiLing Community Site.* http://multiling.iit.demokritos.gr/.
20. Nenkova, A., and McKeown, K. 2012. A survey of text summarization techniques. In *Mining Text Data*, 43-76. Springer US.
21. Ouyang, Y., Li, W., Li, S., and Lu, Q. 2011. Applying regression models to query-focused multi-document summarization. *Information Processing & Management* 47 (2): 227-237.
22. Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. 2002. BLEU: a method for automatic evaluation of machine translation. *40th Annual Meeting of the Association for Computational Linguistics.* 311-318.
23. Scrucca, L. 2013. GA: A Package for Genetic Algorithms in R. *Journal of Statistical Software* 53 (4): 1-37.
24. Therneau, T., Atkinson, B., and Ripley, B. 2015. *rpart: Recursive Partitioning and Regression Trees.* http://CRAN.R-project.org/package=rpart.

Prediction of Happy Endings in German Novels based on Sentiment Information

Albin Zehe, Martin Becker, Lena Hettinger, Andreas Hotho, Isabella Reger, and Fotis Jannidis

University of Würzburg, 97074 Würzburg, Germany
{zehe,becker,hettinger,hotho}@informatik.uni-wuerzburg.de,
{isabella.reger,fotis.jannidis}@uni-wuerzburg.de

Abstract Identifying plot structure in novels is a valuable step towards automatic processing of literary corpora. We present an approach to classify novels as either having a happy ending or not. To achieve this, we use features based on different sentiment lexica as input for an SVM-classifier, which yields an average F1-score of about 73%.

1 Introduction

Every child knows that stories are supposed to have a happy ending. Every adult knows that this is not always true. In fact, in the course of the 19th Century a happy ending became a sign of popular literature, while high literature was marked by a preference for the opposite. This makes happy endings an interesting point of research in the field of digital literary studies, since automatically recognizing a happy ending, as one major plot element, could help to better understand plot structures as a whole.

To achieve this, we need a representation of plot that a computer can work with. In digital literature studies, it has been proposed to use emotional arousal as a proxy [5]. But can we just use existing data mining methods in combination with sentiment features and expect good results for happy ending classification?

In this work, we tackle the problem of identifying novels with a "happy ending" by training a classifier. Our goal is not to present the best method for doing so, but to show that it is generally possible. We introduce our proposed approach, which already yields results considerably above a random baseline, and point out some problems and their possible solutions. Our method uses sentiment lexica in order to derive features with respect to semantic polarity and basic emotions. To account for the structural dynamics of happy endings, these features are built by considering the relation of different sections of the novels. We are able to train a support vector machine (SVM) which yields an average F1 score of 0.73 on a corpus with over 200 labelled German novels. To the best of our knowledge, our work is the first to cover happy ending classification.

The remainder of this paper is structured as follows: related work and background information is presented in Sections 2 and 3. The features and data we use are described in Sections 4 and 5. Then we present our results (Section 6).

In: P. Cellier, T. Charnois, A. Hotho, S. Matwin, M.-F. Moens, Y. Toussaint (Eds.): Proceedings of DMNLP, Workshop at ECML/PKDD, Riva del Garda, Italy, 2016.

2 Related Work

Recently, a lot of attention has been paid to sentiment analysis in the Digital Humanities community. In this section we cover publications constructing features that are useful to our task, but have not actually been used to recognize happy endings.

Matthew Jockers proposed, in a series of blog posts, to use the "analysis of the sentiment markers" as *a novel method for detecting plot* [5, 7, 8]. The basic idea of representing plot by emotions was well received, but the following discussion showed his approach to use Fourier Transformation (FT) and a low-pass filter to smooth the resulting curves is not reasonable, since FT assumes periodicity of the signal [6, 14].

Elsner constructs a representation of the plot in a story using sentiment values, among other features, in [2]. He cites other works stating that sentiment is a very important part of plot development and is therefore critical to automatic understanding of plot.

Mohammad builds emotional representations like ours in [9]. In [1], similar representations are used to automatically compose music from written text.

In [3], Goyal et al. present AESOP, a system that can identify *plot units*. AESOP is partially based on affect states, which are closely related to sentiments.

3 Background

We refer to a novel as having a "happy ending", if the situation of the main characters in the novel improves towards the end of the story or is constantly favourable. In this paper, we propose a method for automatically predicting whether novels have a happy ending or not, based on features derived from sentiment analysis. We start by formally defining the task of "happy ending classification" and introduce some concepts of sentiment analysis which are relevant for our features.

Happy ending classification. We formally define "happy ending classification" as a simple classification task: Given a corpus C, we aim to learn a function $f : C \rightarrow \{0, 1\}$, where $f(c) = 1$ iff a novel $c \in C$ has a "happy ending". In this work, we use a support vector machine (SVM) to train and test the classification function f based on a labelled gold standard. The SVM model requires a feature vector for each novel (cf. Section 5). We mostly use sentiment based features as introduced in Section 4.

Sentiment analysis. Since plot construction, and in particular happy endings, are tightly coupled with sentiments [2], sentiment analysis provides a solid basis for our classification. The goal of sentiment analysis is to determine the *polarity* and *emotions* a human reader would associate with a given word, sentence or other element of a text. In this work, we focus on word-level sentiment analysis.

Polarity denotes if a word has a positive (e.g. friend) or negative (e.g. war) connotation. It can be expressed as a ternary value (-1, 0, or 1). A word can also be associated with a set of *basic emotions*. There are many definitions for basic

emotions, as discussed in [10]. Plutchik et al. define a set of eight basic emotions in [12]: joy, trust, fear, surprise, sadness, disgust, anger and anticipation.

Generally, polarities and emotions are collected in sentiment lexica. Each lexicon contains a set of words which it associates with a number of *sentiment values* according to a set of dimensions (such as polarity or different emotions). In Section 4 we derive different features for each novel based on such a lexicon and in Section 5 we introduce the sentiment lexicon we use in our study.

4 Features

For "happy ending classification" we derive feature vectors based on a set of text *segments* which we combine to form *sections*. The final feature vectors are derived based on certain characteristic values of these segments and sections (e.g. the polarity of the final segment or the difference between the polarity of the first and the last section).

Negation detection. Our features are based on sentiments. However, sentiments can be negated (e.g. "not happy"). For considering negations, we apply the relatively simplistic technique presented in [11]: we add a negation marker to any word between a negation word and the following punctuation, inverting its sentiment score. Following the textblob implementation,[1] we multiply negated sentiments by 0.5, improving results slightly.

Segments. Given a corpus of novels $\mathcal{C}$, we first split each novel $C \in \mathcal{C}$ into n segments, $C = \{S_1, \ldots, S_n\}$. We evenly split by word count resulting in segments of size $\frac{\|C\|}{n}$, where $\|C\|$ denotes the number of words in novel C. Note that the last segment may be shorter due to the length of the novel and the number of segments.[2]

Sentiment values for segments. Now we derive a set of characteristic values for each segment. Given a fixed lexicon L with several dimensions (e.g., the polarity or an emotion), let $v_d(w)$ denote the value lexicon L associates with word w according to dimension d. For example the word "death" is strongly associated with the dimension "sadness", that is, $v_{\text{sadness}}(\text{"death"}) = 1$.

For each segment S_i and each dimension d in the lexicon L, we calculate the characteristic value $\bar{v}_d(S_i)$ as follows:

$$\bar{v}_d(S_i) = \frac{\sum_{w \in L_i} v_d(w)}{|L_i|} \tag{1}$$

where L_i denotes the words in S_i which are covered by lexicon L.

[1] `https://pypi.python.org/pypi/textblob-de/`. The sentiment analysis in textblob is not fully ported to German and does not include basic emotions, so we did not use it directly.

[2] The novels were split into words using textblob-de. Words were lemmatized iff the lexicon used in the respective experiment contained lemmatized forms.

Sections. We merge consecutive segments into sections. We consider two prominent sections, *main* and *final*. The main section $\mathcal{S}_{\text{main}} = \{S_1, \ldots, S_m\}$ covers the majority (75 up to 98%, depending on the experimental setup) of the segments starting from the beginning. The final section $\mathcal{S}_{\text{final}} = \{S_{m+1}, \ldots, S_n\}$ covers the remaining segments and represents the "ending" of the novel. Additionally, we consider a third section, the *late-main* section $\mathcal{S}_{\text{late}} = \{S_{2m-n+1}, \ldots, S_m\}$, which covers the last part of $\mathcal{S}_{\text{main}}$. This section is introduced in order to better capture the sentiment development at the end of the novel. For example, there may be a catastrophic event shortly before the end which is then resolved, leading to a happy ending. Since, in our experiments, the late and the final section are always of the same length, all sections are defined by specifying the number of segments in the main section m.

Sentiment values for sections. For each of these sections, we calculate the characteristic value averages based on the covered segments. In particular, given the segments of a novel, $C = \{S_1, \ldots, S_n\}$, and a section $\mathcal{S}$, we calculate the average characteristic value by extending $\bar{v}_d$ to sections:

$$\bar{v}_d(\mathcal{S}) = \frac{\sum_{S_i \in \mathcal{S}} \bar{v}_d(S_i)}{|\mathcal{S}|} \tag{2}$$

Features. Based on these characteristic values, we finally define the features for each novel. Given n segments, a main section of size m, and a lexicon L, the feature vector contains the following values for each dimension d: (1) the characteristic value of the final section $f_{d,\text{final}} = \bar{v}_d(\mathcal{S}_{\text{final}})$, (2) the characteristic value of the last segment $f_{d,\text{n}} = \bar{v}_d(S_n)$, (3) the difference between the main and the final section $f_{d,\text{main-final}} = \bar{v}_d(\mathcal{S}_{\text{main}}) - \bar{v}_d(\mathcal{S}_{\text{final}})$ and (4) the difference between the late-main and the final section $f_{d,\text{late-final}} = \bar{v}_d(\mathcal{S}_{\text{late}}) - \bar{v}_d(\mathcal{S}_{\text{final}})$. The change in sentiment values towards the end of the novel is characterized by the two differences. The difference was used to ensure that generally sad novels that had a significant improvement in the final segments can still be classified as having a happy ending or, in reverse, a drop in positive emotions towards the end of a generally happy novel can be recognized as a sad ending.

5 Dataset

In this section, we describe our annotated corpus, as well as the sentiment lexicon we derive our features from.

Annotated novels. Our dataset consists of 212 German novels compiled from the TextGrid Digital Library[3] and the Projekt Gutenberg[4], mostly written between 1750 and 1920. The number of words in the novels ranges from less than 20,000 words up to more than 300,000. These novels have been manually labelled

[3] `https://textgrid.de/digitale-bibliothek`
[4] `http://gutenberg.spiegel.de`

Table 1. Examples for entries in the NRC lexicon

Word	pos.	neg.	anger	antic.	disg.	fear	joy	sadn.	surp.	trust
Entführung	0	1	1	0	0	1	0	1	1	0
verachten	0	1	1	0	1	0	0	0	0	0
Bewunderung	1	0	0	0	0	0	1	0	0	1

by domain experts as either having a happy ending or not,[5] based on sources like the Kindler,[6] Wikipedia[7] or by reading relevant parts of the novel. Half of the novels (106) are annotated as having a happy ending. The annotated data can be made available upon request.

Sentiment lexica. In this work, we employ the German version of the NRC sentiment lexicon,[8] which is provided by the original author of the English version [10]. It encompasses the following semantic dimensions: if a word is positive (0 or 1) or negative (0 or 1), and if a word is associated with some basic emotion (each 0 or 1). We also add another dimension, i.e., the "polarity", which is the negative value subtracted from the positive value.

After removing duplicates and all-zero entries, which would not help in our task, the lexicon contains 4597 entries, as exemplified in Table 1. We also evaluated our approach on SentiWS [13] (polarity scores $\in \{-1, 0-1\}$) and GPC (German Polarity Clues) [15] (polarity scores $\in [-1, 1]$), however, achieving inferior results.

6 Results and Discussion

In this section we train a support vector machine (SVM)[9] for classifying happy endings of novels on the annotated corpus introduced in Section 5 using the features presented in Section 4. For the SVM we use an RBF kernel and the parameters $C = 1$ and $\gamma = 0.01$. A linear kernel gave slightly worse results, grid search for parameter selection did not lead to an improvement. We standardize the features using the sklearn StandardScaler[10] before classification. All tests were run with 10-fold cross-validation.

Baselines. Since our dataset is equally divided in novels with and without a happy ending, the random baseline as well as the ZeroR classifier (which assigns every novel to the largest class) reach 50% accuracy. Dropping the notion of sections, and only using the average sentiment values of the entire novel (i.e.,

[5] Because this is a simple task, each novel was labelled by only a single domain expert.
[6] `http://www.derkindler.de`
[7] `de.wikipedia.org`
[8] `http://saifmohammad.com/WebPages/NRC-Emotion-Lexicon.htm`
[9] We tried some other classifiers as well, with Random Forests and Naive Bayes reaching about the same score, while k-NN and Decision Trees performed worse.
[10] `http://scikit-learn.org/stable/modules/preprocessing.html`

Table 2. Results for an overall segment count of $n = 75$, a main section containing $m = 71$ segments, and the NRC lexicon averaged over 20 iterations, each using 10-fold cross validation.

happy ending	precision	recall	f1-score	support
False	0.72	0.73	0.73	106
True	0.73	0.72	0.73	106
avg/**total**	0.73	0.73	0.73	**212**

one value for each dimension in L) yields an F1 score of 0.54, which is slightly above the random baseline. Adding the scores of the last segment improves the F1 score to about 0.66 for the best performing segment count $n = 75$. This suggests that the final segment of the novel is indeed an important feature for classifying happy endings.

Best parameter configuration. Our method requires to choose a sentiment lexicon and the number of segments n, as well as the length of the main section m (in segments). We compared different configurations and found that working with the NRC as introduced in Section 5 using $n = 75$ segments with a main section of $m = 71$ segments worked best. Other lexica containing only polarity scores (cf. Section 5) performed worse, suggesting that the combination of basic emotions represents a more accurate picture of the overall mood in a novel than polarity alone. Table 2 shows the results with the best configuration, accumulated over 20 iterations.[11]

Influence of segmentation and section size. In this paragraph, we describe how changing the number of segments n and the percentage of segments assigned to the main section $\mathcal{S}_{\text{main}}$, that is $\frac{|\mathcal{S}_{\text{main}}|}{n}$, influences the results. Figure 1 shows the average F1-score over 20 test runs based on the NRC lexicon. Each line corresponds to a segment count n. From a larger set of segment counts we chose the 4 best performing ones. The x-axis corresponds to the percentage of segments in the main section. The y-axis shows the F1-score achieved with the respective configuration. It can be seen that splitting the novels into 50 segments mostly works very well, but is outperformed by 75 segments with a main section of 71 segments (about 95%). Furthermore, most segmentations perform best when using about 5% or 10% of the segments as the final section.

Limitations. Here, we list some limitations of our work and suggest possible solutions.

One variation we tried was to limit our features to sentences containing explicit references to the main character of the novel. The intuition behind this is that the concept of happy ending is closely related to the fate of protagonists of a story. Using a domain-adapted named entity recognition (NER) toolkit [4], we selected the main character as the one being explicitly named most often. After

[11] The results vary slightly between iterations, with total F1-scores mostly between 71% and 75%. Averaging over 20 iterations yields stable results.

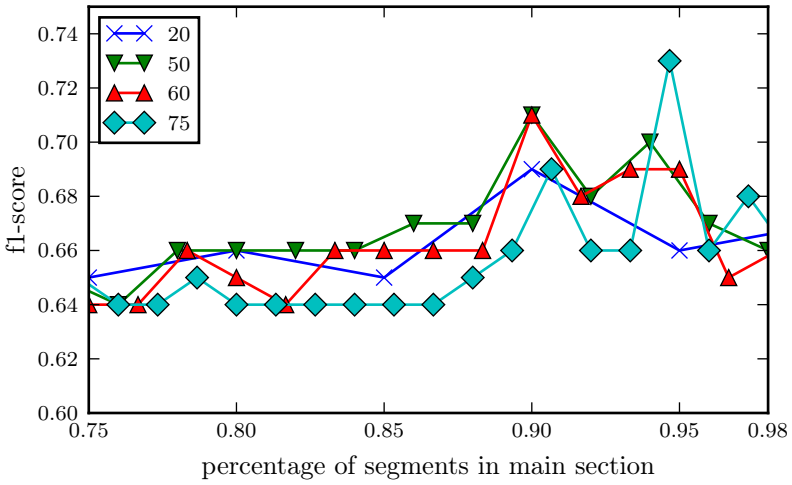

Figure 1. Plot of different segmentation configurations. Segment count n is represented as different lines, the x-axis corresponds to the number of segments in the main section m and the y-axis is the F1-score.

segmenting the novel, we then removed all sentences not mentioning this character. Contrary to our expectation, this did not improve results but indeed led to a significant drop in accuracy. The reason for this might be our decision to (for now) avoid error-prone co-reference resolution or our strict choice of focusing only on a single main character.

Employing more sophisticated sentiment analysis would likely improve our results. For example, while our relatively crude negation detection only led to slight improvements, considering a more advanced set of sentiment shifters should help to get better results.

We also did not take into account that some stories are not told in chronological order. Those stories are difficult to our system, as the happy ending may happen at some arbitrary point in the text. Working around this problem would require a way to identify corresponding scenes in different novels.

Finally, we are currently working on a way to choose the length of the main section individually for each novel, instead of passing it to the model as a fixed hyperparameter.

7 Conclusion

In this work, we have presented an SVM classifier for identifying novels with happy endings. Our approach is based on features derived from sentiment lexica and exploits structural dynamics by comparing different sections of the novels.

We consider the F1-score of 0.73 to be a good starting point for future work, such as evaluating our method on more extensive labelled datasets. Additionally, it is interesting to investigate if the same parameters yield good results for different novel collections, or different languages.

References

1. Davis, H., Mohammad, S.: Generating music from literature. In: Proceedings of the 3rd Workshop on Computational Linguistics for Literature (CLFL). pp. 1–10. Association for Computational Linguistics, Gothenburg, Sweden (April 2014)
2. Elsner, M.: Abstract representations of plot struture. LiLT (Linguistic Issues in Language Technology) 12 (2015)
3. Goyal, A., Riloff, E., Daumé III, H.: Automatically producing plot unit representations for narrative text. In: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing. pp. 77–86. Association for Computational Linguistics (2010)
4. Jannidis, F., Krug, M., Puppe, F., Toepfer, M., Weimer, L., Reger, I.: Automatische Erkennung von Figuren in deutschsprachigen Romanen (2015)
5. Jockers, M.L.: A novel method for detecting plot (Jun 2014), `http://www.matthewjockers.net/2014/06/05/a-novel-method-for-detecting-plot/`
6. Jockers, M.L.: Requiem for a low pass filter (Apr 2015), `http://www.matthewjockers.net/2015/04/06/epilogue/`
7. Jockers, M.L.: The rest of the story (Feb 2015), `http://www.matthewjockers.net/2015/02/25/the-rest-of-the-story/`
8. Jockers, M.L.: Revealing sentiment and plot arcs with the syuzhet package (Feb 2015), `http://www.matthewjockers.net/2015/02/02/syuzhet/`
9. Mohammad, S.: From once upon a time to happily ever after: Tracking emotions in novels and fairy tales. In: Proceedings of the 5th ACL-HLT Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities. pp. 105–114. Association for Computational Linguistics, Stroudsburg, PA, USA (2011)
10. Mohammad, S.M., Turney, P.D.: Crowdsourcing a word-emotion association lexicon 29(3), 436–465 (2013)
11. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up?: Sentiment classification using machine learning techniques. In: Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10. pp. 79–86. Association for Computational Linguistics (2002)
12. Plutchik, R.: A general psychoevolutionary theory of emotion. Theories of emotion 1, 3–31 (1980)
13. Remus, R., Quasthoff, U., Heyer, G.: Sentiws – a publicly available german-language resource for sentiment analysis. In: Proceedings of the 7th International Language Resources and Evaluation (LREC'10). pp. 1168–1171 (2010)
14. Schmidt, B.M.: Commodius vici of recirculation: the real problem with syuzhet (Apr 2015), `http://benschmidt.org/2015/04/03/commodius-vici-of-recirculation-the-real-problem-with-syuzhet/`
15. Waltinger, U.: Sentiment analysis reloaded: A comparative study on sentiment polarity identification combining machine learning and subjectivity features. In: Proceedings of the 6th International Conference on Web Information Systems and Technologies (WEBIST '10). Valencia, Spain (April 2010)

A Study of Synthetic Oversampling for Twitter Imbalanced Sentiment Analysis

Julien Ah-Pine and Edmundo Pavel Soriano Morales

University of Lyon, ERIC Lab
5 avenue Pierre Mendès France
69676 Bron Cedex, France
{julien.ah-pine,edmundo.soriano-morales}@univ-lyon2.fr

Abstract. The majority of Twitter sentiment analysis systems implicitly assume that the class distribution is balanced while in practice it is usually skewed. We argue that Twitter opinion mining using learning methods should be addressed in the framework of imbalanced learning. In this work, we present a study of synthetic oversampling techniques for tweet-polarity classification. The experiments we conducted on three publicly available datasets show that these methods can improve the recognition of the minority class as well as the geometric mean criterion.

Key words: Synthetic sampling, Sentiment analysis, Social media.

1 Introduction

Micro-blogging services are communication tools that are massively used by people to instantaneously share their opinions about any kinds of topics. These opinions are of interest for companies or individuals, like politicians, as they allow them to monitor their online reputation. Twitter has been the most popular micro-blogging service with more than 500 million tweets per day in 2013[1]. Thus, sentiment analysis of tweets[2] has received a lot of attention both from academia and industry during the last years.

In this paper, we focus on tweets polarity classification using supervised learning methods. This task is challenging in several respects. Firstly, tweets are limited to 140 characters and they contain irregular lexical units and syntactic patterns. Hence, these data are noisy, sparse and high-dimensional which makes the learning process difficult. Moreover, tweets expressing an opinion about a given topic usually present a skewed polarity distribution. In this case, any classifier would be biased towards the majority class.

In order to cope with these challenges, we propose to use synthetic oversampling techniques. These procedures are designed to deal with the class imbalance issue. We show that not only they enable reducing the bias towards the majority

[1] http://www.internetlivestats.com/twitter-statistics/
[2] Short informal messages in a more general perspective.

In: P. Cellier, T. Charnois, A. Hotho, S. Matwin, M.-F. Moens, Y. Toussaint (Eds.): Proceedings of DMNLP, Workshop at ECML/PKDD, Riva del Garda, Italy, 2016.

class, but they also alleviate the data sparsity burden commonly encountered in text mining.

The rest of the paper is organized as follows. In section 2, we discuss some related works in order to position and motivate our proposal. In section 3, we present our approach based on three synthetic oversampling methods and two supervised learning methods. Then, in section 4, we detail the experiments we conducted on three datasets including two different languages and we discuss the obtained results as well. We conclude the paper in section 5.

2 Related Works

2.1 Twitter Sentiment Analysis

Twitter sentiment analysis has received a growing interest starting from 2009 [5, 19]. In this work, we focus on polarity detection which aims at predicting the opinion of a tweet as positive or negative. Supervised learning techniques are the mainstream approaches in this case. Due to the characteristics of Twitter data, systems usually used for sentiment analysis (see [14] for a survey of this field) do not perform well. In order to improve classifiers' performance for tweets opinion mining, most of research works have proposed to extract features/lexicons which are specific to this type of data and/or leverage external resources [5, 19, 11, 22, 10, 20, 15]. In contrast, we apply a corpus-based approach with no particular feature engineering.

2.2 Imbalanced Sentiment Analysis

The class imbalance problem in binary classification occurs when the sizes of the classes differ greatly. In this case, any classifier is biased toward the majority class (see [9] for a survey of the domain). For example, in the datasets we examined, near 70% of the tweets of the datasets we experimented with are negative. If a naïve classifier always assigns the negative polarity to any tweet, it will give an overall accuracy of 70% but without recovering any positive tweet, which is not satisfying.

Imbalanced learning for sentiment analysis has been studied by several researchers in different learning settings [12, 13, 17, 25]. However, we found very few papers that directly address imbalanced sentiment analysis for Twitter data [16, 6]. The methods that are proposed in the two latter works are similar to cost-sensitive approaches. In our case, we rather use sampling techniques.

3 The Proposed Approach

3.1 Vector Space Representation and Neighborhood

Tweets contain slang words and irregular expressions. Thus, linguistic analyses by conventional NLP tools often give poor performances on such texts. To

circumvent these difficulties, and also to deal with different languages, we rely on a vectorial representation of tweets based on a bag-of-words approach. We denote by $\mathcal{F}$ the resulting feature space, $\mathbf{x} \in \mathcal{F}$ is a vector representing a tweet and its coordinates are its words' frequency. In what follows, we use $\mathbb{P}$ and $\mathbb{N}$ to designate the subsets of tweets with the minority and the majority class labels respectively ($|\mathbb{P}| < |\mathbb{N}|$).

In order to compare tweets, we use the cosine similarity function. Note that all pairwise proximity measures lie between 0 and 1 since the coordinates of vectors are non-negative. Let $\mathbf{x}$ be any tweet in $\mathbb{P}$ then its neighborhood is denoted $\mathbb{NN}(\mathbf{x})$ and it consists of the k nearest neighbors.

3.2 Synthetic Oversampling

To face the skewed class distribution problem, one straightforward approach is to balance the training set so that $|\mathbb{P}| = |\mathbb{N}|$. Undersampling the majority class or oversampling the minority class are two possible strategies. Since the data are very sparse, undersampling the majority class is sub-optimal as we may lose meaningful examples in the learning process. Therefore, oversampling the minority class seems a better solution. In this case, synthetic oversampling creates new examples in $\mathbb{P}$ by taking convex combinations of existing points.

We recall three popular synthetic oversampling methods: SMOTE [2], Borderline-SMOTE [7] and ADASYN [8]. Their general procedure can be cast as follows:

1. Select an original tweet $\mathbf{x}$ according to a probability distribution over $\mathbb{P}$.
2. Determine $\mathbb{NN}(\mathbf{x})$.
3. Select a neighbor $\mathbf{x}'$ according to a probability distribution over $\mathbb{NN}(\mathbf{x})$.
4. Create a synthetic example $\mathbf{y}$ as follows:

$$\mathbf{y} = \mathbf{x} + \alpha(\mathbf{x}' - \mathbf{x}) \tag{1}$$

   where α is a random value in $[0, 1]$.
5. Repeat 1-4 until the desired number of new examples is reached.
6. Append the set of synthetic points to $\mathbb{P}$.

Note that $\mathbf{y}$ lies in the line segment joining $\mathbf{x}$ and $\mathbf{x}'$. It is important to notice that $\mathbf{y}$ belongs to the subspace spanned by the union of the underlying subspaces of $\mathbf{x}$ and $\mathbf{x}'$. Therefore, synthetic examples are less sparse than original ones.

The main differences between the three oversampling methods concern the random selection of $\mathbf{x} \in \mathbb{P}$ in step 1. SMOTE assumes a uniform distribution over $\mathbb{P}$ whereas Borderline-SMOTE assumes a uniform distribution over $\mathbb{B}$, a subset of $\mathbb{P}$. $\mathbb{B}$ consists of tweets in $\mathbb{P}$ whose neighborhoods contain a majority of points in $\mathbb{N}$. These items lie in subspaces where the decision boundary is prone to errors. Thereby, it is expected that oversampling in these parts of the space improves the classifier performances. Regarding ADASYN, it assumes a non uniform distribution over $\mathbb{P}$. It can be seen as a smoothed version of Borderline-SMOTE: the noisier the neighborhood of $\mathbf{x}$, the more synthetic points around $\mathbf{x}$. In other words, the probability to select $\mathbf{x}$ in step 1 is proportional to the number of points of $\mathbb{N}$ contained in $\mathbb{NN}(\mathbf{x})$.

4 Experiments

4.1 Datasets and Data Representation

We assess the approach introduced previously on three publicly available Twitter datasets. The first two are OMD "Obama-McCain Debate" [21] and HCR "Health Care Reform" [23]. The third one is IW "Imagiweb" and concerns tweets in French, posted during the 2012 french presidential election [24]. We chose political tweets because they present a particularly skewed class label distribution.

Concerning the vectorial representation of tweets, we used unigrams of words and we only removed the hapax.

We give the descriptive statistics[3] of these datasets below:

- OMD: 1906 tweets (710 positive, 1196 negative) and 1569 features;
- HCR: 1922 tweets (541 positive, 1381 negative) and 2066 features;
- IW: 4519 tweets (1092 positive, 3427 negative) and 3918 features.

4.2 Supervised Learning Methods

We experimented with two different learning models: decision trees and the $l1$ penalized logistic regression.

Decision trees are well-known symbolic learning techniques and offer the advantages of coping with high-dimensional data as well as providing human-readable outputs. In this work, we used CART [1], which builds a binary classification tree based on the Gini index splitting criterion. The R package `rpart` was used and the default parameters values specified in `rpart.control` were applied.

The $l1$ penalized logistic regression [18] is also an appropriate supervised learning for high-dimensional data since it implicitly performs feature selection. Moreover, this method has proven to provide competitive results in text classification [4]. We used the `glmnet` R package [3] and in particular the function `cv.glmnet` which allows us to select the mixing parameter λ based on the error observed during training phase.

4.3 Assessment Measures

We use several performance criteria: overall accuracy (OA), F1-measures of the positive and negative classes (F-$\mathbb{P}$ and F-$\mathbb{N}$ respectively). OA evaluates the overall performance of a classifier but it does not properly account for the performances on $\mathbb{P}$ as compared to $\mathbb{N}$ because of the skewed distribution of class labels. Hence, we also use a popular criterion for imbalanced learning: the geometric mean (GM) of both class accuracy rates. Unlike OA, GM is independent of the class distribution (see [9, Chapter 8] for an overview of this topic). Thus we argue that GM should also be a default evaluation criterion in Twitter sentiment analysis tasks.

[3] We removed tweets that were labeled as neutral since we are only concerned with polarity detection.

4.4 Experiments Setting and Results

It is important to note that we are not interested in comparing the results of decision trees against $l1$ penalized logistic regression. Our purpose is rather to illustrate that synthetic oversampling can improve the performances of learning methods on Twitter imbalanced-polarity detection tasks.

We tested the two learning models on the three collections with different relatively balanced training sets. In what follows, τ is a variable taking its values in $\{0, 1/4, 1/2, 3/4, 1\}$ which measures how much the training set is balanced with respect to the initial distribution. In fact, $\tau = 0$ is when no oversampling was carried out and we used the initial imbalanced training set (this is our baseline); $\tau = 1/4$ means we generated $\lfloor(|\mathbb{N}| - |\mathbb{P}|)/4\rfloor$ positive synthetic examples; ...; and $\tau = 1$ means we exactly sampled $|\mathbb{N}| - |\mathbb{P}|$ new positive items in order to have a perfectly balanced training set. The neighborhood was set to $k = 20$ nearest neighbors[4]. The results we obtained using a 5 fold cross-validation are plotted in Figure 1 for decision trees and in Figure 2 for $l1$ penalized logistic regression.

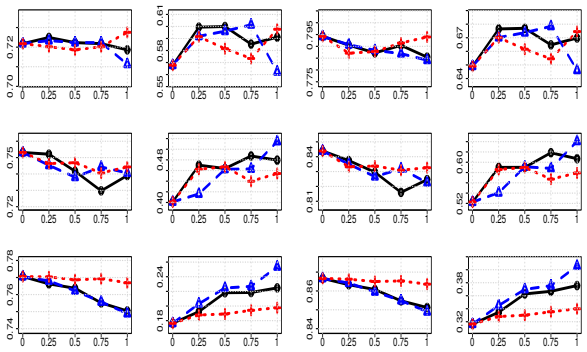

Fig. 1: Results for decision trees (CART). Solid line with circles refers to SMOTE, dashed line with triangles refers to Borderline-SMOTE and dotted line with plus signs refers to ADASYN. From left to right: plots of OA, F-$\mathbb{P}$, F-$\mathbb{N}$ and GM measures. From top to bottom: plots for OMD, HCR and IW benchmarks. The x-axis refers to τ going from initial imbalanced ($\tau = 0$) to fully balanced ($\tau = 1$) training sets.

Our main findings are the following:

– For both decision tree and $l1$ penalized logistic regression, we note quite the same trends: oversampling generally improves the results. Indeed, All

[4] We also tested with $k = 10, 30$ but the trends were similar and the results comparable.

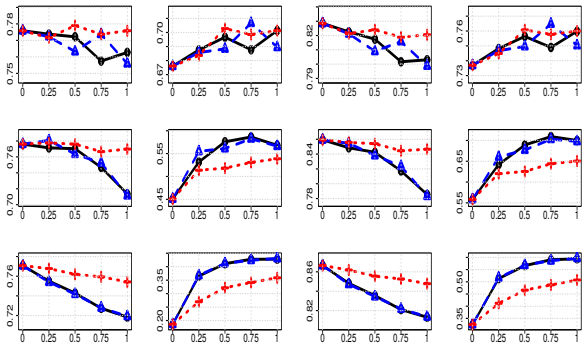

Fig. 2: Results for $l1$ penalized logistic regression. Same legend as in Figure 1.

three sampling methods globally improve the GM measure[5]. Thereby, our approach allows alleviating the class imbalance problem effectively. For OMD, when $\tau = 1$, the most important gains for GM measures are given by ADASYN (1st row, 4th column in the figures). Regarding HCR and IW, Borderline-SMOTE performs the best but SMOTE often provides comparable results (2nd and 3rd rows respectively and 4th column in the figures).

- All three oversampling strategies generally boosts F-$\mathbb{P}$ values[6]. The minority class is thus better recognized. However, this is at the expense of a reduction of F-$\mathbb{N}$ values. Nonetheless, since the increasing rate of F-$\mathbb{P}$ is generally much larger than the decreasing rate of F-$\mathbb{N}$, we note the overall increase of GM values as highlighted previously.
- For all three sampling techniques, the OA measure tends to diminish as the training set is more and more balanced. In fact, since the class distribution in the test set is skewed towards $\mathbb{N}$, the errors on true negative tweets have more impact on OA than the correct detection of true positive tweets. This illustrates again the fact that OA is not a criterion that properly accounts for imbalanced data.
- We cannot conclude on which of the three oversampling strategies is the best. However, we can make the following remarks:
  - SMOTE and Borderline-SMOTE have quite the same behaviours for the HCR and IW collections. F-$\mathbb{P}$ measures are greater than ADASYN whereas F-$\mathbb{N}$ values are lower. Both methods allows a much better recognition of the minority class but in doing so they make more mistakes when detecting the majority class.
  - In contrast, ADASYN presents peculiar properties. The increase of GM values are lower than for the two other methods but this oversampling

[5] The only exception is observed for OMD when using a fully balanced training sets ($\tau = 1$) generated by Borderline-SMOTE with CART as shown in Figure 1.

[6] Except the same particular case mentioned previously.

technique shows more stable OA values and even better ones in some cases. For the OMD dataset specifically, this approach not only provides among the best performances for the GM criterion but it also allows improving the OA measures unlike the other methods.

5 Conclusion

Twitter sentiment analysis is confronted with the class imbalance problem and it is important to take this aspect into account when designing opinion mining systems based on machine learning.

A way to address this challenge is to use synthetic oversampling which aims at balancing the training set in a meaningful way. Three state-of-the-art methods have been examined in that regard. We conducted experiments on political-tweets polarity classification using three datasets and in two different languages. The obtained results show that our proposal makes it possible to deal with the skewed class distribution issue by providing better recognition of the minority class as well as obtaining large increases of the overall geometric mean criterion.

In future work, we intend to extend our study to multiclass sentiment analysis and also to examine the use of synthetic oversampling methods in other NLP tasks as a general approach to cope with the sparsity problem.

Acknowledgment This work was partly supported by the french national projects Imagiweb ANR-2012-CORD-002-01 and Request PIA/FSN.

References

1. Breiman, L., Friedman, J., Stone, C.J., Olshen, R.A.: Classification and regression trees. CRC press (1984)
2. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: Synthetic minority over-sampling technique. J. Artif. Int. Res. 16(1) (2002)
3. Friedman, J., Hastie, T., Tibshirani, R.: Regularization paths for generalized linear models via coordinate descent. Journal of Statistical Software 33(1) (2010)
4. Genkin, A., Lewis, D.D., Madigan, D.: Large-scale bayesian logistic regression for text categorization. Technometrics 49 (2007)
5. Go, A., Bhayani, R., Huang, L.: Twitter Sentiment Classification using Distant Supervision. Tech. rep., Stanford University (2009), `https://sites.google.com/site/twittersentimenthelp/home`
6. Hamdan, H., Bellot, P., Bechet, F.: Lsislif: Feature extraction and label weighting for sentiment analysis in twitter. In: Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015) (2015)
7. Han, H., Wang, W.Y., Mao, B.H.: Borderline-smote: A new over-sampling method in imbalanced data sets learning. In: Advances in Intelligent Computing, Lecture Notes in Computer Science, vol. 3644 (2005)
8. He, H., Bai, Y., Garcia, E., Li, S.: Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In: Neural Networks, 2008. IJCNN 2008. (IEEE World Congress on Computational Intelligence) (2008)

9. He, H., Ma, Y.: Imbalanced Learning: Foundations, Algorithms, and Applications. Wiley-IEEE Press (2013)
10. Kiritchenko, S., Zhu, X., Mohammad, S.M.: Sentiment analysis of short informal texts. Journal of Artificial Intelligence Research (JAIR) 50 (2014)
11. Kouloumpis, E., Wilson, T., Moore, J.D.: Twitter sentiment analysis: The good the bad and the omg! In: Proceedings of the Fifth International Conference on Weblogs and Social Media, Barcelona, Catalonia, Spain, July 17-21, 2011 (2011)
12. Li, S., Wang, Z., Zhou, G., Lee, S.Y.M.: Semi-supervised learning for imbalanced sentiment classification. In: Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume Three. IJCAI'11 (2011)
13. Li, S., Zhou, G., Wang, Z., Lee, S.Y.M., Wang, R.: Imbalanced sentiment classification. In: Proceedings of the 20th ACM International Conference on Information and Knowledge Management. CIKM '11 (2011)
14. Liu, B., Zhang, L.: A survey of opinion mining and sentiment analysis. In: Aggarwal, C.C., Zhai, C. (eds.) Mining Text Data. Springer US (2012)
15. Martínez-Cámara, E., Martín-Valdivia, M.T., Urena-López, L.A., Montejo-Ráez, A.R.: Sentiment analysis in twitter. Natural Language Engineering 20(01) (2014)
16. Miura, Y., Sakaki, S., Hattori, K., Ohkuma, T.: Teamx: A sentiment analyzer with enhanced lexicon mapping and weighting scheme for unbalanced data. In: Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014) (2014)
17. Mountassir, A., Benbrahim, H., Berrada, I.: An empirical study to address the problem of unbalanced data sets in sentiment classification. In: Systems, Man, and Cybernetics (SMC), IEEE International Conference on (2012)
18. Ng, A.Y.: Feature selection, l1 vs. l2 regularization, and rotational invariance. In: Proceedings of the Twenty-first International Conference on Machine Learning. ICML '04 (2004)
19. Pak, A., Paroubek, P.: Twitter as a corpus for sentiment analysis and opinion mining. In: LREC. vol. 10 (2010)
20. Saif, H., He, Y., Fernandez, M., Alani, H.: Semantic patterns for sentiment analysis of twitter. In: Proceedings of the 13th International Semantic Web Conference - Part II. ISWC '14 (2014)
21. Shamma, D.A., Kennedy, L., Churchill, E.F.: Tweet the debates: Understanding community annotation of uncollected sources. In: Proceedings of the First SIGMM Workshop on Social Media. WSM '09 (2009)
22. Sidorov, G., Miranda-Jiménez, S., Viveros-Jiménez, F., Gelbukh, A., Castro-Sánchez, N., Velásquez, F., Díaz-Rangel, I., Suárez-Guerra, S., Treviño, A., Gordon, J.: Empirical study of machine learning based approach for opinion mining in tweets. In: Advances in Artificial Intelligence (2012)
23. Speriosu, M., Sudan, N., Upadhyay, S., Baldridge, J.: Twitter polarity classification with label propagation over lexical links and the follower graph. In: Proceedings of the First Workshop on Unsupervised Learning in NLP. EMNLP'11 (2011)
24. Velcin, J., Kim, Y., Brun, C., Dormagen, J., SanJuan, E., Khouas, L., Peradotto, A., Bonnevay, S., Roux, C., Boyadjian, J., Molina, A., Neihouser, M.: Investigating the image of entities in social media: Dataset design and first results. In: Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014) (2014)
25. Xu, R., Chen, T., Xia, Y., Lu, Q., Liu, B., Wang, X.: Word embedding composition for data imbalances in sentiment and emotion classification. Cognitive Computation 7(2) (2015)

Shallow Text Clustering Does Not Mean Weak Topics: How Topic Identification Can Leverage Bigram Features

Julien Velcin[1], Mathieu Roche[2], and Pascal Poncelet[3]

[1] Université de Lyon (ERIC, Lyon 2), France.
Julien.Velcin@univ-lyon2.fr,
[2] Cirad (TETIS), Montpellier, France.
mathieu.roche@cirad.fr
[3] Université de Montpellier (LIRMM), France.
Pascal.Poncelet@lirmm.fr

Abstract. Text clustering and topic learning are two closely related tasks. In this paper, we show that the topics can be learnt without the absolute need of an exact categorization. In particular, the experiments performed on two real case studies with a vocabulary based on bigram features lead to extracting readable topics that cover most of the documents. Precision at 10 is up to 74% for a dataset of scientific abstracts with 10,000 features, which is 4% less than when using unigrams only but provides more interpretable topics.

1 Introduction

Text clustering is a huge research area with many applications, such as corpus visualization [12] and document indexing for information retrieval [25]. In addition to the classical task of categorizing similar texts, people are usually interested in characterizing the clusters by the mean of concise descriptions called topics, so that they can easily interpret categories and browse the document collection [24]. Topic extraction (or topic learning) has been widely popularized by the success of Latent Semantic Analysis [4] and Non-negative Matrix Factorization [18]. More recently, probabilistic topic models, such as probabilistic Latent Semantic Analysis [8] and Latent Dirichlet Allocation [1], have emerged as an efficient alternative implemented by many communities, from data mining [19] to natural language processing [7] and social sciences and humanities [21]. They are now used as a routine in many systems dedicated to text analytics [2].

Despite all these numerous works, it turns out that some confusion often subsists between the task of **text clustering** (grouping similar texts, i.e. working on category's *extension*) and the task of **topic identification** (extracting within-category commonalities, their *intension*), as highlighted by [26]. We show here that not-so-good (shallow) clustering does not always mean weak topics. Another observation is related to the vocabulary used by the algorithms: most of the time, groups and topics are estimated from unigram tokens (words) [17], whose number is often arbitrarily fixed, or not fully justified [7]. When considering perplexity-based measures only, that is the goodness-of-fit of the probabilistic model on held-out data, words seems to play the main role [9]. However, it has been shown that n-grams ($n \geq 2$) might be really useful, whether for constructing interpretable topics [23] or for improving topic consistency [16,28].

In: P. Cellier, T. Charnois, A. Hotho, S. Matwin, M.-F. Moens, Y. Toussaint (Eds.): Proceedings of DMNLP, Workshop at ECML/PKDD, Riva del Garda, Italy, 2016.

Based on these two observations, our contribution is twofold.

First, we show that a minimum number of features is necessary but sufficient to achieve a good accuracy, both in term of clustering purity and topic description. It is not as obvious as it seems since too many features might add noise and reduce the generalization ability of the model, which actually happens in supervised settings [13]. If we pay attention to select enough features, it is therefore possible to choose phrases (here, bigrams) instead of single words. To the best of our knowledge, it is the first time that this result is clearly highlighted and quantified.

Second, we show that the bigram-based vocabulary provide really useful topic descriptions at the cost of a reasonable decrease in accuracy. The cost is not that important with a drop of about 10%. Our results highlight that a careful choice for the features allows a much better interpretation of the topics given by topic learning techniques (here, LDA). This work is closely related to the task of topic labeling but, here, the descriptive features are defined *before* the topic learning step. Therefore the extracted topics are characterized by the very terms that constitute their backbone, and not labeled by using one among the many heuristics proposed in the literature [15,29]. Besides, a post-processing can be used afterwards to improve the output, such as selecting one term amongst "data set" and "data sets" (see Section 3).

The paper is organized as follows. Section 2 defines the two complementary tasks of text clustering and topic identification, highlighting their close connection but also their difference. Section 3 shows the impact of making the vocabulary change in term of both size and nature (unigrams versus bigrams). Finally, we conclude and suggest future work in Section 4.

2 Text Clustering and Topic Identification: Two Related Tasks

2.1 Definition of Tasks

The first step consists in showing the clear distinction between the two tasks. As illustrated in Fig. 1 (left), text clustering mainly aims at categorizing objects into clearly separated clusters. Even though the membership can be gradual, like in fuzzy clustering [5], or an object can be associated to several clusters, like in overlapping clustering [3], the common aim is to associate each object to one category so that subsequent decisions can be made. In Fig. 1, we can observe that some texts are central to the categories (e.g., d_c for cluster 1 and d_d for cluster 2) whereas other texts lie between clusters (e.g., d_a, d_b and d_e). It is a natural feature of text clustering to assume that texts can be related to *several* topics at the same time, which is at the basis of most topic models.

By adopting a different viewpoint, topic identification is more dedicated to extracting a set of topics that structure the dataset as shown in Fig. 1 (right). Topics can be viewed as weighted lists of keywords (e.g., with LSA) or distributions over words (e.g., with LDA). In order to give an overview of the whole corpus to the final users, the usual solution is to keep the top words (option 1 in Fig. 1) or the top n-grams (option 2 in Fig. 1, here with n=2).

Obviously, the two tasks are related but not fully aligned. Hence, the documents d_a, d_b and d_e can be misclassified as long as we find the expected topics, more identifiable

on the colored groups of documents in Fig. 1. Let us note that we might easily get the top frequent terms for each cluster as a post-processing stage. However, most of the current state-of-the-art methods such as LSA, NMF and LDA address both tasks at the same time, which explains the confusion that may arise.

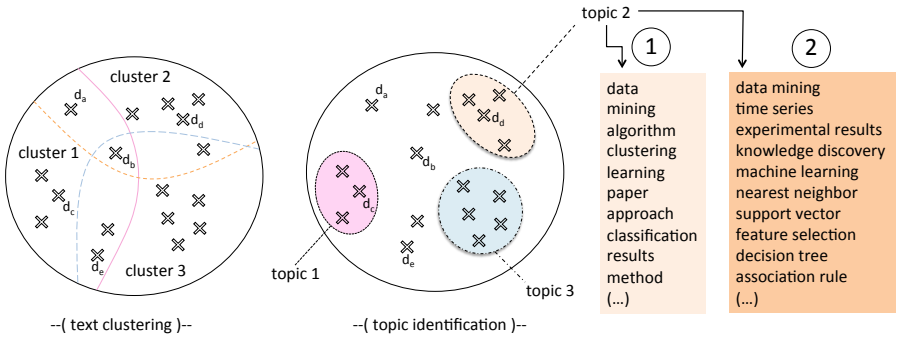

Fig. 1. Distinction between the tasks of text clustering and topic identification (here, topic 2 has been extracted from scientific publications clearly related to the "data mining" field).

Several previous works have used n-grams either during the topic learning process [22,23] or as a post-processing step in order to find automatic labels [10,15,29]. However, they did not study the impact of both the vocabulary size (number of terms) and term nature (unigrams versus bigrams), as we do in this paper.

2.2 Evaluation Measures

In the following sections, we experiment LDA on two datasets in order to address the two tasks simultaneously. For evaluation's sake, we compare the output given by LDA to a gold standard that provides us the real class label of each object. In order to get a partition, we associate each text d to the most likely topic $\hat{z} = \arg\max p(z/d)$. This way, we reduce the expressive power of topic models but we can leverage the usual Adjusted Rand Index (ARI) for assessing the clustering quality. We will see in the next section that the two datasets have been precisely chosen because they fit this crisp clustering assumption. The maximum of 1 with ARI is achieved with a perfect match between the partition and the gold standard.

Establishing a ground truth for the topic identification task is much more challenging. To begin with, we choose to restrict the evaluation to the quality of the top-10 terms associated to each cluster for 10 is the number usually shown to end users. Although we can imagine various ways to extract those lists from the gold standard partition (e.g., selecting the most discriminant terms, etc.), we have chosen to restrict to the most frequent terms for this study. In addition to the simplicity of this solution, we will see that the probability $p(t/z)$ of the term t given the topic z output by LDA clearly favor frequent terms. We then propose to calculate the usual precision for this top-10 terms, noted pre@10. Let us remark that this manner to challenge the list of top K terms is

especially uncommon in the literature, in which the list is always manually evaluated. The mapping between the true category and the topic z is simply derived by taking the category with the higher number of texts related to z. Obviously, pre@10 ranges from 0 (no common term) to 1 (perfect match between the two lists).

2.3 Datasets and Feature Extraction

dataset	#c	#docs	#unigrams	#bigrams
ART	5	18,465	13,778	30,522
20NG	20	18,828	40,142	54,741

Fig. 2. Basic statistics for the two datasets.

The two datasets are the set of scientific abstracts gathered by Tang. et al. [20], noted ART, and 20 Newsgroups, noted 20NG. Both datasets are available online[4]. For both datasets, we perform minimal preprocessing: lowercasing, removing punctuation and English stopwords, removing the terms that occur in only one document. We set the number of expected topics to be the true number of classes #c in the gold standard. Basic statistics can be found in Fig. 2.

In our context, we extract bigrams based on classical patterns in terminology extraction domain (i.e. noun-noun, adjective-noun, and so forth)[5]. Terms extracted from our corpora are then ranked depending on their relative frequency. Other weightings have been experimented (e.g., *TF-IDF*, *Okapi*, *C-value*) but it turns out that the frequency is the more adapted ranking function for both tasks addressed in this study[6].

3 Vocabulary Impact for Both Tasks

We here focus our attention on the importance of vocabulary size and term nature (unigrams versus bigrams). We used the parallel LDA implemented in the MALLET package[7]. The priors α and β are not automatically estimated (default configuration) but we set them both to 0.1 after a preliminary grid search[8]. We set the maximum number of iterations for the Gibb's sampling to 2000, as suggested with this implementation, and run the algorithm ten times. The final mean is only given since the observed standard deviation does not exceed 0.01, so we decided not to overload the figures.

We keep the most frequent K words, K ranging from 500 to 30,000. We then compute the quality of LDA topics both for clustering (Fig. 3) and topic identification

[4] http://arnetminer.org/collaboration and http://qwone.com/ jason/20Newsgroups/

[5] To this end, we used the biotex tool [11], freely available online: http://tubo.lirmm.fr/biotex/.

[6] For instance, the ARI based on the frequency is higher from 0.25 to 0.31 for ART and a vocabulary of 10,000 features.

[7] Homepage of MALLET package: http://mallet.cs.umass.edu

[8] It turns out that, with this amount of data, priors had a limited effect on the final results (± 0.015 on ARI). We are aware that an automatic, dynamic estimation is possible [14] but we do believe that a constant setup of hyperparameters guarantees a fair comparison.

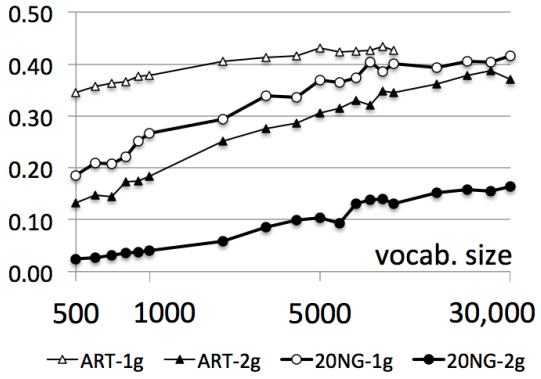

Fig. 3. Evolution of ARI (log scale for x axis).

(Fig. 4). We first observe that ARI increases exponentially below some threshold before converging[9]. This means that a fraction of features is sufficient to get an important gain in ARI (5 000 unigrams for ART achieves 0.434 for a maximum of 0.4346 with 9 000 unigrams ; 10,000 unigrams for 20NG achieves 0.3961 for a maximum of 0.4285 with 30,000 unigrams). For information, recent work [6,27] focusing on text clustering reported 0.397 and 0.425 ARI on 20NG respectively.

In addition, we observe that with three times the number of features, bigram-based vocabulary is able to achieve a really good ARI score for ART, not very far from the maximum with unigrams (0.3865 against 0.4346). This is clearly not the same situation for 20NG with a difference of about 0.26 for the ARI.

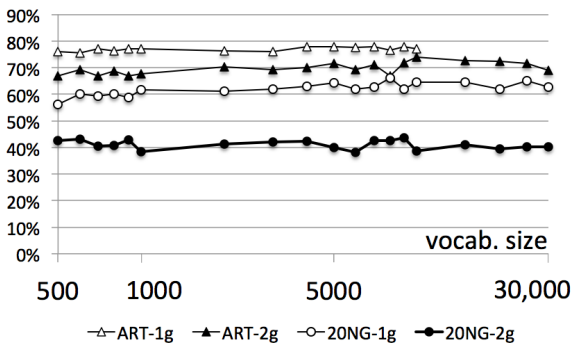

Fig. 4. Evolution of prec@10 (log scale for x axis).

We now take a closer look at the top terms returned by LDA, in comparison to the reference terms extracted from the true classes. The precision achieved by keeping the top-10 terms is shown in Fig. 4. We have been really surprised to notice that the

[9] We stopped the size for ART-1g at the number of words occurring at least twice in the whole corpus (13,778 words).

datamining (ART)		sci.space (20NG)		rec.sport.baseball (20NG)	
1-grams	2-grams	1-grams	2-grams	1-grams	2-grams
data	data mining*	space	solar system*	writes	red sox*
mining	data sets*	earth	henry spencer*	game	san francisco (15)
algorithm	association rules*	launch	physical universe (30)	article	los angeles (18)
clustering	time series*	writes	night sky*	year	st louis*
paper	data streams (13)	shuttle	space shuttle*	team	world series*
approach	experimental results*	nasa	toronto zoology*	games	major league*
learning	knowledge discovery*	mission	oort cloud (12)	good	blue jays*
classification	data set*	orbit	jet propulsion*	players	power play
algorithms	machine learning (12)	system	dick dunn (25)	baseball	mark singer*
results	support vector*	solar	high-speed collision (26)	time	san diego (12)

Fig. 5. top-10 terms of selected topics for ART (columns on the left) and 20NG (columns on the right). * means that the bigram is in the top-10 bigrams extracted from the ground truth, otherwise we note its rank. Henry Spencer posted over 34,000 messages to the sci.space.* newsgroups (source: Wikipedia).

top-10 bigrams are really competitive in comparison to the top-10 words. For ART, the precision achieved is 4% to 7% below only with a score of about 70% (up to 74% for 10,000 terms). We believe that this is a crucial observation: even though we get one term less that with single words in average, the topics are much more readable by using seven bigrams than height unigrams. The bigram "data mining" is more informative than the two words "data" and "mining", even if they are given in the same list.

This advantage is obvious if we take a look at the top terms given in the table of Fig. 5 (top bigrams next to top unigrams). For 20NG, it is even more interesting: despite the ARI collapse, four of the top bigrams are still accurate (six for the unigrams). However, it does not mean that the other terms are unrelated. We have highlighted the terms related to the ground truth with a * in Fig. 5, and noted their rank in the true list otherwise. Let us note that a vocabulary of 500 terms is sufficient to achieve such performances in term of precision. It seems that LDA easily finds the core of topics, without caring much for the result of the text clustering task.

Finally, we have run a last series of experiments in order to see the impact of a mixed unigram-bigram vocabulary. To this end, we fixed the number of features to 10,000 and changed the proportion in steps of 5% (e.g., 80% unigrams with 20% bigrams). The results confirm that bigrams might help increasing the overall clustering accuracy, but the bonus is limited and not significant. The best proportion seems to be highly dependent of the dataset (e.g., we got +0.01 ARI for ART with 5% of bigrams and +0.024 for 20NG with 30%). However, we observed no constant improvement for the precision. When we take a closer look to the top terms, bigrams are overwhelmed by unigrams, which explains the unchanged score.

4 Discussion and Future Work

Despite all the work done so far for integrating phrases into topic learning, we believe that this study is the first to highlight the potentiality of bigrams, not only for improving topic homogeneity (in addition to unigrams) or topic labeling, but for the whole task of

topic identification. Even though we have observed a clear gap between unigram and bigram frequencies, the bigram frequency seems to be sufficient to cover most of topic's aspects, getting rid of the ambiguity carried by unigrams. Hence, it is easy to provide readable topics to end users with a limited energy in the creation of terms (actually, any bigram library is expected to provide interesting features). Our preliminary experiments have shown that this reasoning can be transposed to trigrams as soon as their cumulated frequency is sufficient. We observed a decrease of about 10% for the pre@10 with trigrams (60% for ART and 30% for 20NG).

Interesting work lies ahead. One immediate follow-up is to design a new method that directly focuses on topic identification. By even more weakening our expectations on text clustering, we can find a way to improve the top K terms by favoring the topical core of each category (colored areas in Fig. 1). Improving the input representation, for instance by adding pseudo-counts for complex terms, can be a way to explore this idea. Another exciting, more theoretical question is to question the tradeoff between term frequency, term co-occurrences and performances. During our experiments, we observed a clear logarithmic correlation between the total number of tokens and the performances we can achieve (from $R^2 = 0.94$ for 20NG until $R^2 = 0.98$ for ART described with bigrams). This tells us that we cannot expect much by using too rare terms since they lead to really sparse matrices. However, it seems that the combination of complementary rare terms can compete with more frequent words. Information theory might be used for studying this kind of issues further.

References

1. David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3:993–1022, 2003.
2. Richard T Carback III, Brad D Gaynor, Nathan R Shnidman, and Sang Hoon Chin. Systems and methods for software analytics, December 17 2015. US Patent 20,150,363,197.
3. Guillaume Cleuziou. An extended version of the k-means method for overlapping clustering. In *Proceedings of the 19th International Conference on Pattern Recognition*, 2008.
4. Scott C. Deerwester, Susan T Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. Indexing by latent semantic analysis. *JASIS*, 41(6):391–407, 1990.
5. Joseph C. Dunn. Well-separated clusters and optimal fuzzy partitions. *Journal of cybernetics*, 4(1):95–104, 1974.
6. Siddharth Gopal and Yiming Yang. Von mises-fisher clustering models. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 154–162, 2014.
7. David Hall, Daniel Jurafsky, and Christopher D Manning. Studying the history of ideas using topic models. In *Proceedings of the conference on empirical methods in natural language processing*, pages 363–371. Association for Computational Linguistics, 2008.
8. Thomas Hofmann. Probabilistic latent semantic analysis. In *Proceedings of the 15th conference on Uncertainty in Artificial Intelligence*, pages 289–296. Morgan Kaufmann, 1999.
9. Jey Han Lau, Timothy Baldwin, and David Newman. On collocations and topic models. *ACM Trans. Speech Lang. Process.*, 10(3):10:1–10:14, July 2013.
10. Jey Han Lau, Karl Grieser, David Newman, and Timothy Baldwin. Automatic labelling of topic models. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 1536–1545. Association for Computational Linguistics, 2011.
11. Juan Antonio Lossio-Ventura, Clement Jonquet, Mathieu Roche, and Maguelonne Teisseire. BIOTEX: A system for biomedical terminology extraction, ranking, and validation. In *Proceedings of the ISWC 2014 - the 13th International Semantic Web Conference, ISWC 2014, Riva del Garda, Italy, October 21, 2014.*, pages 157–160, 2014.

12. Dongning Luo, Jing Yang, Milos Krstajic, William Ribarsky, and Daniel Keim. Eventriver: Visually exploring text collections with temporal references. *IEEE Transactions on Visualization and Computer Graphics*, 18(1):93–105, 2012.
13. Christopher D Manning, Prabhakar Raghavan, Hinrich Schütze, et al. *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge, 2008.
14. Andrew Mccallum, David M Mimno, and Hanna M Wallach. Rethinking lda: why priors matter. In *Advances in Neural Information Processing Systems*, pages 1973–1981, 2009.
15. Qiaozhu Mei, Xuehua Shen, and ChengXiang Zhai. Automatic labeling of multinomial topic models. In *13th ACM SIGKDD international conference on Knowledge Discovery and Data mining*, pages 490–499, 2007.
16. Michael Nokel and Natalia Loukachevitch. A Method of Accounting Bigrams in Topic Models. In *Proceedings of the 11th Workshop on Multiword Expressions*, pages 1–9, Denver, Colorado, 2015. Association for Computational Linguistics.
17. Derek O'Callaghan, Derek Greene, Joe Carthy, and Pádraig Cunningham. An analysis of the coherence of descriptors in topic modeling. *Expert Systems with Applications*, 42(13), 2015.
18. Pentti Paatero and Unto Tapper. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5(2), 1994.
19. Mark Steyvers, Padhraic Smyth, Michal Rosen-Zvi, and Thomas Griffiths. Probabilistic author-topic models for information discovery. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge Discovery and Data mining*, pages 306–315, 2004.
20. Jie Tang, Sen Wu, Jimeng Sun, and Hang Su. Cross-domain collaboration recommendation. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1285–1293, 2012.
21. Christoph Wagner, Vera Liao, Peter Pirolli, Lynn Nelson, and Markus Strohmaier. It's not in their tweets: Modeling topical expertise of twitter users. In *Privacy, Security, Risk and Trust (PASSAT), collocated with the IEEE international conference on Social Computing (SocialCom)*, pages 91–100, 2012.
22. Hanna M Wallach. Topic modeling: beyond bag-of-words. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 977–984. ACM, 2006.
23. Xuerui Wang, Andrew McCallum, and Xing Wei. Topical n-grams: Phrase and topic discovery, with an application to information retrieval. In *Proceedings of the 7th IEEE International Conference on Data Mining (ICDM), 2007*, pages 697–702, 2007.
24. Furu Wei, Shixia Liu, Yangqiu Song, Shimei Pan, Michelle X. Zhou, Weihong Qian, Lei Shi, Li Tan, and Qiang Zhang. TIARA: a visual exploratory text analytic system. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, July 25-28, 2010*, pages 153–162, 2010.
25. Weili Wu, Hui Xiong, and Shashi Shekhar. *Clustering and information retrieval*, volume 11. Springer Science & Business Media, 2013.
26. Pengtao Xie and Eric P Xing. Integrating document clustering and topic modeling. In *Proceedings of the 29th Conference on Uncertainty in Artificial Intelligence (UAI)*, 2013.
27. Xiaohui Yan, Jiafeng Guo, Yanyan Lan, and Xueqi Cheng. A biterm topic model for short texts. In *Proceedings of the 22nd international conference on World Wide Web (WWW)*, pages 1445–1456. ACM, 2013.
28. Yi Zhang, Guangquan Zhang, Hongshu Chen, Alan L. Porter, Donghua Zhu, and Jie Lu. Topic analysis and forecasting for science, technology and innovation: Methodology with a case study focusing on big data research. *Technological Forecasting and Social Change*, 2016.
29. Wayne Xin Zhao, Jing Jiang, Jing He, Yang Song, Palakorn Achananuparp, Ee-Peng Lim, and Xiaoming Li. Topical keyphrase extraction from twitter. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT)*, pages 379–388, 2011.

Topic Models with Sparse and Group-Sparsity Inducing Priors

Christian Pölitz

TU Dortmund University, Otto Hahn Str. 12, 44227 Dortmund

Abstract. The quality of topic models highly depends on quality of used documents. Insufficient information may result in topics that are difficult to interpret or evaluate. Including external data to can help to increase the quality of topic models. We propose sparsity and grouped sparsity inducing priors on the meta parameters of word topic probabilities in fully Bayesian Latent Dirichlet Allocation (LDA). This enables controlled integration of information about words.

1 Introduction

Topic models have been used for text analysis in the last decade very successfully. Topic models assign a number of latent topics to documents and words from a given corpus. These topics can be interpreted as different meanings of words or semantic clusters of the documents in a text corpus. In text analysis the topics can be used in many ways.

The estimation of the topics highly depends on the amount of text data used. Considering the case when we have only very limited amounts of texts to estimate a topic model, the quality of the found topics can be quite poor. In such situation external information about the words can be quite beneficial. For instance prior word probabilities can help sampling word topic distributions from a Dirichlet distribution by adding prior weights on more likely words. In this sense, we try to align the topics with an external probability model like a language model $p(w)$ over some of the words. Structural external information like similarities of words can provide further help to align the topics. Hence, prior weights of whole groups of similar words can be used to estimate the topics.

To measure the quality of the found topics, intrinsic measures like the perplexity have been used in the past. Recently, coherence measures have been introduced as an evaluation measure for topics that agree well with human judgements, see [11]. These coherence measures use external information to evaluate how much related the most likeliest words in the topics are. To extract coherent topics by a topic model we must assume to have enough coherent documents. This is not always the case. In Word Sense Induction for instance, there may be rare words that appear only in a few documents. In such a case these documents might not be enough to generate coherent topics. Further, very sparse documents as in collections of Blog posts or Tweets might also lack enough information to extract coherent topics.

To increase the coherence, we propose to integrate external information like word probabilities or word similarities from external data sources. To control the influence from the external information we weight these information additionally. We integrate

In: P. Cellier, T. Charnois, A. Hotho, S. Matwin, M.-F. Moens, Y. Toussaint (Eds.): Proceedings of DMNLP, Workshop at ECML/PKDD, Riva del Garda, Italy, 2016.

external word probability information by appropriate prior distributions. We add a sparsity prior and a group sparsity prior on the log-likelihood of the topic model, see [16]. The sparsity inducing priors can now actively control the amount and the weight of the external information to be integrated in the estimation of a topic model. From the group sparsity we expect more coherence since whole groups of words are considered. These groups are expected to be more coherent since they are similar based on some external information.

2 Related Work

There are many previous approaches integrating external information into the generation of a topic model. [8] use a regression model on the hyperparameters of the Dirichlet prior for LDA. They use Dirichlet multinomial regression to make the prior probability of the document topic distribution dependent on document features. [14] integrate word features into LDA by adding a Logistic prior on the parameter of the Dirichlet prior of the word topic distribution. [9] integrate correlation information about words into a topic model. They propose regularized topic models that have structural priors instead of Dirichlet priors. These structural priors contain word co-occurrence statistics for instance. [7] propose a Pólya Urn Model to integrate co-occurrence statistics into a topic model. [4] use First Order Logic incorporated into LDA to leverage domain knowledge. [3] incorporate information about words that should or should not be together in a topic from topic model. [6] integrate lexical semantic relations like synonyms or antonyms derived from external dictionaries into a topic model.

In the last years many approaches have been proposed to evaluate topic models. [17] propose to estimate the probability of some held-out documents of the collection used for topic modelling. The authors propose several sampling techniques to efficiently approximate this probability. [10] propose to evaluate topic models based on external information. They use pointwise mutual information (PMI) based on co-occurrence statistics from external text sources to evaluate topics. [11] evaluate topic models by coherence of the topics. They authors showed that the coherence measure agrees with human evaluations of the topics. [1] evaluate topics based on distributional semantics. They find semantic spaces such that words that are semantically related based on statistics on Wikipedia are close in these spaces. [15] developed a framework for measuring coherence in topics. The authors performed large empirical experiments on standard data sets and possible coherence measures to evaluate the framework.

3 Topic Models with Prior Information of Words

We integrate external information into LDA via priors on the word-topic distributions. Similar to the approach by [8], we define an asymmetric Dirichlet prior with metaparameter β on the word topic distribution θ. β specifies the prior believe on the distribution of the words before we have seen any data. We make β dependent on the word distribution from the external information $p(w)$. We set $\beta_{w,t} = exp(\lambda_{w,t}) \cdot p(w)$ for a weight parameter $\lambda_{w,t}$ for the individual influence of the prior information in each topic. If $\lambda_{w,t}$ is zero, the prior believe of the probability of w is directly used. If $\lambda_{w,t}$ is

less than zero, the prior believe is weighted down. If $\lambda_{w,t}$ is greater than zero, the prior believe is weighted up.

The optimal parameters λ must be found by optimizing the likelihood of the topic model. We perform alternating optimization of the parameters with quasi Newton methods and Gibbs sampling of topics to find the optimal topic model.

For the optimization of the parameters we minimize the part of the negative log likelihood from standard LDA that depends on β:

$$L = \sum_t \log \Gamma(\tilde{\beta}_t + n_k) - \log \Gamma(\tilde{\beta}_t) + \sum_t \sum_{w:n_{w,t}>0} \log \Gamma(\beta_{w,t}) - \log \Gamma(\beta_{w,t} + n_{w,t})$$

with $\tilde{\beta}_t = \sum_w \beta_{w,t}$.

3.1 Sparsity Priors for LDA

We propose to use a sparsity inducting priors on the parameter $\lambda_{w,t}$ weights to influence the prior information about word w for topic t. We expect that some parts of the prior information play a bigger role than other parts in the estimated topic model. To find out which parts are important we impose sparsity to identify them.

We add a Laplace prior on the λ parameters to gain sparsity. This means, we aim at reducing the amount of adaptation of the external information. This has three advantages. First, we can easily read from the parameters which parts of the prior information influences the topics. Second, we get a simpler model that adapts the external prior information only for some words. Third, we gain control on the amount of external information to be integrated into the topic model.

The difference to standard LDA is that we have now an asymmetric prior β that is derived from the external information (the word probabilities) and the weight of this information has a Laplace prior. Adding the Laplace prior of the λ parameters of the DMR and optimizing for the negative log-likelihood is the same as putting a sparsity inducing penalty on them. Now, the negative log likelihood is simply extended by $\|\lambda_t\|_1$:

$$L_1 = L + \sigma^{-1} \sum_t \|\lambda_t\|_1.$$

Hence, the Laplace prior is integrated into the optimization via a sparse lasso penalty $\|\lambda\|_1$. We solve the optimization problem via Orthantwise Quasi Newton Optimization [2].

3.2 Group-Sparsity Priors for LDA

The previous idea of limiting the adaptation of the external prior information for some words does not consider that the information about similar words should also be treated similar. For instance, in case the prior information about the word “book” is not adapted, we should also not adapt the information about “author” or “books”.

We propose to add a group lasso penalty to the negative log likelihood to gain group sparsity:

$$L_2 = L + \sigma^{-1}\|\lambda\|_1 + \sum_g \gamma^{-1}\|\lambda_g\|_2$$

for the group lasso penalty $\sum_g \gamma^{-1} \cdot \|\lambda_g\|_2$ for the groups g and the variance γ. Conceptionally, this is the same as having a prior on the λ parameters that induces group sparsity.

Similar to above we solve the group lasso via Blockwise Coordinate Descent with Proximal Operators for the group penalty, see [5] for more details.

3.3 Finding Groups

To find the groups for the grouped sparsity priors on the weight parameters we use external information about similarities of words. From such similarities we can easily generate clusters that are used as groups. We divide the weight parameter $\lambda = (\lambda_1, \cdots, \lambda_G)$ with G partial weights $\lambda_g = (\lambda_{w_1,g}, \cdots \lambda_{w_k,g})$. The partial weights build a group g if the words $w_1, \cdots w_k$ build a cluster based on the similarities from the external information. The similarities we use are based on WordNet (see [13]). We generate a so called affinity matrix M such that $(M)_{ij} = exp(-(1 - sim(w_i, w_j))$ for sim the similarity derived from WordNet. Next, we perform a spectral clustering [12] to find the groups. Spectral clusterings performs a simple k-means clusterings on the words projected onto low-dimensional space spanned by the eigenvectors of the affinity matrix.

4 Experiments

In this section, we investigate the topics extracted by our proposed methods (**SparsePrior**) for LDA with sparsity prior, (**GroupPrior**) for LDA with group sparsity prior) and compare them with two standard state-of-the-art implementations of topic models that integrate external information about words: (**RegLDA**) by [9] and (**WordFeatures**) by [14]. Additionally, we also compare to the standard LDA with Gibbs sampling without external information. For each method, we use $T = 20$ topics, 1000 iterations and set $\alpha = 50/T$, $\beta = 0.1$ (for standard LDA and topic models with structural prior), $\gamma^{-1} = 0.1$, $\sigma^{-1} = 0.1$.

4.1 Data sets

We use two standard text data sets used in previous approaches of topic modelling. First, we use the 20 Newsgroups[1] data set. The data set contains about 20.000 text documents from 20 different newsgroups. Overall we have 1000 documents per newsgroup. We additional remove stop words and prune very infrequent and very frequent words. Second, we use the Senseval-3[2] data set of English lexical samples. The data set contains

[1] http://qwone.com/ jason/20Newsgroups/

[2] http://www.senseval.org/senseval3

Data	20 newsgroups				Wikipedia			
Method	NPMI	UCI	UMASS	nLL	NPMI	UCI	UMASS	nLL
LDA	-0.065	-2.268	-5.250	2332131	-0.065	-2.268	-5.250	2332131
WordFeatures	-0.061	-2.135	-4.825	**2330149**	-0.061	-2.135	-4.825	**2330149**
RegLDA	-0.069	-2.443	-5.520	2332699	-0.069	-2.443	-5.520	2332699
SparePrior	-0.070	-2.472	-5.359	2334633	-0.070	-2.472	-5.359	2334633
GroupPrior	**-0.055**	**-2.116**	**-4.796**	2333298	**-0.055**	**-2.116**	**-4.796**	2333298

Table 1. Results on the different data sets: 20 newsgroups data set and Wikipedia talk pages.

Data	SensEval			
Method	NPMI	UCI	UMASS	MI
LDA	-0.050	-1.712	-3.706	0.359
WordFeatures	-0.058	-1.744	-4.096	0.328
RegLDA	-0.056	-1.767	-3.693	0.323
SparePrior	-0.025	-0.747	-3.060	0.290
GroupPrior	**-0.021**	**-0.634**	**-3.056**	**0.360**

Table 2. Results on the SensEval data set.

texts from Penn Treebank II Wall Street Journal article. The sizes of the data sets range from 20 to 200 documents per word. Further, we use the wikipedia talk pages to apply the method to a more recent data source of internet based communication. As example, we extract 10.000 postings of discussions on wikipedia from 2002 to 2014 that contain the term ”cloud”.

4.2 Coherence Results

In the first experiments, we compare to the state-of-the-art LDA implementations with external information about words and standard LDA in terms of quality. We want to show that our model produces more coherent topics. To evaluate the coherence of

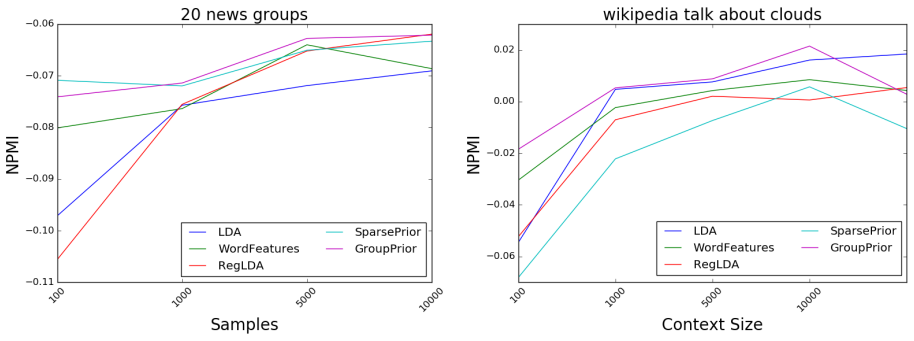

Fig. 1. NPMI for different sample sizes and document length used.

the found topics, we use Pointwise Mutual Information (UCI), normalized Pointwise Mutual Information (NPMI) and arithmetic mean of conditional probability (UMass), see [15]. Further, for the two larger data sets 20 news groups and the postings from wikipedia we also estimate the negative log-likelihood (nLL) on a held out data set. Finally, on the SensEval data set, we also estimate the Mutual Information (MI) of the found topics to the true sense.

The results on the 20 newsgroups data set on the left in Table 1 show that our proposed group sparsity prior results in topic with better coherence measures than the state-of-the art methods and the standard LDA. From the state-of-the-art competitors only **WordFeatures** performs comparably good. In terms of loglikelihood, **WordFeatures** performs best. For the wikipedia talk pages we get similar results as shown on the middle in Table 1.

Finally, we compare the different topic model methods on collection of very small data sets. In Table 2 shows the resulting coherence values on the SensEval data set. LDA with our proposed grouped sparsity prior performs better on all data samples compared to the competitor.

We are especially interested in how the different methods perform on very small data sets. To investigate this, we evaluate the NPMI for the different methods on different sample sizes and different document lengths of the samples. For the 20 news groups date, we sample 100, 100, 5000 and 10000 documents to extract topics. From the wikipedia talk pages we extract postings of different context sizes from 100 to 1000 characters. In Figure 1, we see that our propose sparsity and group sparsity priors results for small samples and small context sizes in the highest NMPI. In these situations our proposed methods of using the group sparsity pays of the most.

5 Conclusion

In this paper we propose to integrate external information about words into topic models to increase topic coherence. We use different priors on the metaparameters for LDA. To control the amount of the integration of the external information we weight them individually. Adding sparsity inducing priors on these weights enables active control on the how much we adapt the external information. By this we trade off topic coherences and likelihood of the topics. Our proposed group sparsity prior further enables integration of external similarity information about words. Now, we can influence the external information of whole groups of words that are similar. The results on large data collections showed the benefit of our proposed method in terms of topic coherence. Finally, we showed that on very small data sets, the group sparsity inducing prior results in better performance.

References

1. Nikolaos Aletras and Mark Stevenson. Evaluating topic coherence using distributional semantics. In *Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013) – Long Papers*, pages 13–22, Potsdam, Germany, March 2013. Association for Computational Linguistics.
2. Galen Andrew and Jianfeng Gao. Scalable training of l1-regularized log-linear models. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, pages 33–40, New York, NY, USA, 2007. ACM.
3. David Andrzejewski, Xiaojin Zhu, and Mark Craven. Incorporating domain knowledge into topic modeling via dirichlet forest priors. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 25–32, New York, NY, USA, 2009. ACM.
4. David Andrzejewski, Xiaojin Zhu, Mark Craven, and Benjamin Recht. A framework for incorporating general domain knowledge into latent dirichlet allocation using first-order logic. In *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011*, pages 1171–1177, 2011.
5. Francis Bach, Rodolphe Jenatton, and Julien Mairal. *Optimization with Sparsity-Inducing Penalties (Foundations and Trends(R) in Machine Learning)*. Now Publishers Inc., Hanover, MA, USA, 2011.
6. Zhiyuan Chen, Arjun Mukherjee, Bing Liu, Meichun Hsu, Malu Castellanos, and Riddhiman Ghosh. Discovering coherent topics using general knowledge. In *Proceedings of the 22Nd ACM International Conference on Information & Knowledge Management*, CIKM, pages 209–218, New York, NY, USA, 2013. ACM.
7. David Mimno, Hanna M. Wallach, Edmund Talley, Miriam Leenders, and Andrew McCallum. Optimizing semantic coherence in topic models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 262–272, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.
8. David M. Mimno and Andrew McCallum. Topic models conditioned on arbitrary features with dirichlet-multinomial regression. *CoRR*, abs/1206.3278, 2012.
9. David Newman, Edwin V. Bonilla, and Wray L. Buntine. Improving topic coherence with regularized topic models. In John Shawe-Taylor, Richard S. Zemel, Peter L. Bartlett, Fernando C. N. Pereira, and Kilian Q. Weinberger, editors, *NIPS*, pages 496–504, 2011.
10. David Newman, Sarvnaz Karimi, and Lawrence Cavedon. External evaluation of topic models. In *Australasian Document Computing Symposium*, pages 11–18, Sydney, December 2009.
11. David Newman, Jey Han Lau, Karl Grieser, and Timothy Baldwin. Automatic evaluation of topic coherence. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 100–108, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.
12. Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS*, pages 849–856. MIT Press, 2001.
13. Ted Pedersen, Siddharth Patwardhan, and Jason Michelizzi. Wordnet::similarity: Measuring the relatedness of concepts. In *Demonstration Papers at HLT-NAACL 2004*, HLT-NAACL–Demonstrations '04, pages 38–41, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics.
14. James Petterson, Alexander J. Smola, Tibrio S. Caetano, Wray L. Buntine, and Shravan M. Narayanamurthy. Word features for latent dirichlet allocation. In John D. Lafferty, Christopher K. I. Williams, John Shawe-Taylor, Richard S. Zemel, and Aron Culotta, editors, *NIPS*, pages 1921–1929. Curran Associates, Inc., 2010.

15. Michael Röder, Andreas Both, and Alexander Hinneburg. Exploring the space of topic coherence measures. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, WSDM '15, pages 399–408, New York, NY, USA, 2015. ACM.
16. Noah Simon, Jerome Friedman, Trevor Hastie, and Robert Tibshirani. A sparse-group lasso. *Journal of Computational and Graphical Statistics*, 22(2):231–245, 2013.
17. Hanna M. Wallach, Iain Murray, Ruslan Salakhutdinov, and David Mimno. Evaluation methods for topic models. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 1105–1112, New York, NY, USA, 2009. ACM.

Topic Sentiment Joint Model with Word Embeddings

Xianghua Fu, Haiying Wu, Laizhong Cui

College of Computer Science and Software Engineering, Shenzhen University, Shenzhen Guangdong 518060, China

fuxh@szu.edu.cn, whywuhaiying@gmail.com, cuilz@szu.edu.cn

Abstract. Topic sentiment joint model is an extended model which aims to deal with the problem of detecting sentiments and topics simultaneously from online reviews. Most of existing topic sentiment joint modeling algorithms infer resulting distributions from the co-occurrence of words. But when the training corpus is short and small, the resulting distributions might be not very satisfying. In this paper, we propose a novel topic sentiment joint model with word embeddings (TSWE), which introduces word embeddings trained on external large corpus. Furthermore, we implement TSWE with Gibbs sampling algorithms. The experiment results on Chinese and English data sets show that TSWE achieves significant performance in the task of detecting sentiments and topics simultaneously.

1 Introduction

With the rapid development of e-commerce and social media, it is extremely urgent and valuable to automatically analyze the reviews to detect sentiments and topics simultaneously. Great effort on new methodologies for detecting topics and sentiments simultaneously has flourished in the recent years [1-5].

Several works extending probabilistic topic models[6,7] have been designed to tackle the problem of the joint extraction of sentiments and latent topics from documents in the recent years [2, 3, 8]. The joint sentiment topic model (JST) [2] extends LDA to a four-layer model by adding an additional sentiment layer between the document and the topic layers. Topic sentiment mixture (TSM) [8] jointly models topics and sentiments in the corpus built on the basis of PLSI. These approaches infer sentiment and topic distributions from the co-occurrence of words within documents. However, when the training corpus is small or when the documents are short, the sentiment and topic distributions might be not very satisfactory. Additionally, most of recent works [2, 3, 9] try to incorporate some polarity lexicons into their models as the prior knowledge. However, these approaches still have their limitations, for example if the polarity lexicons are not rich, the improvement of the prior is very limited. As a result, we have to seek for other approaches.

Most recently, word embeddings are gaining more and more attention, since they show very good performance in a broad range of natural language processing (NLP) tasks [10-12]. For example, [10] incorporates latent feature vector representations of

In: P. Cellier, T. Charnois, A. Hotho, S. Matwin, M.-F. Moens, Y. Toussaint (Eds.): Proceedings of DMNLP, Workshop at ECML/PKDD, Riva del Garda, Italy, 2016.

words to LDA model, and [11] employs latent topic models to assign topics for each word in the text corpus, and learns topical word embeddings (TWE). But these models only complete the task of mining topics. Little attention has been devoted to topic sentiment model with word embeddings so far. In this paper, we propose a new topic sentiment model which incorporates word embeddings. To the best of our knowledge, it is the first work to formulate topic sentiment model with word embeddings.

In contrast with other topic sentiment modeling frameworks, our model is distinguished from them as follows: (1) we incorporate word embeddings trained on very large corpora. It significantly improves the sentiment-topic-word mapping and extends semantic and syntactic information of words. (2) experiments are performed on four real online review data sets for two kinds of language (English and Chinese), which show that our model is used more extensive. (3) we also compare the performance on incorporating the sentiment polarity and without introducing sentiment polarity respectively to demonstrate that our new model is fully unsupervised. We find that our unsupervised model is highly portable to other domains for the sentiment classification task and achieves significant performance in the task of sentiment analysis, and extracting sentiment-specific topics.

2 Topic and Sentiment Model with Word Embeddings

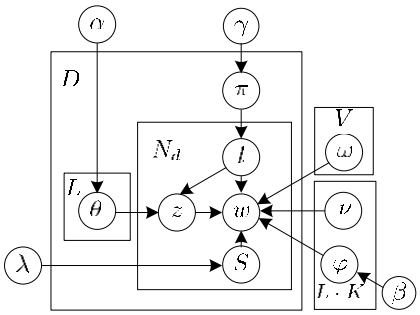

Fig. 1. Graphical representation of TSWE model

2.1 Topic and Sentiment Model with Word Embeddings

In this section, we propose a novel topic sentiment model with word embeddings called TSWE, as shown in Fig. 1. TSWE is formed by taking the original topic sentiment model JST [2, 3] and replacing their Dirichlet multinomial component with a two components mixture of a sentiment-topic-to-word Dirichlet multinomial component and a word embeddings component. Our model defines the probability that it generates a word from embeddings component as the multinomial distribution $MulT$ with:

$$MulT(w_i|\nu_k\omega^T) = \frac{\exp(\nu_k \cdot \omega_{w_i})}{\sum_{w_i' \in W} \exp(\nu_k \cdot \omega_{w_i'})} \quad (1)$$

The negative log likelihood L according to our model factorizes topic-wise into factors L_k for each topic associated with sentiment. we derive:

$$L_k = \mu \parallel \nu_k \parallel_2^2 - \sum_{w_i \in W} N^{k,w_i} \left(\nu_k \omega_{w_i} - \log \left(\sum_{w_i' \in W} \exp \left(\nu_k \omega_{w_i'} \right) \right) \right) \tag{2}$$

Then we apply L-BFGS implementation [13] from the Mallet toolkit [14] to derive the topic vector ν_k that minimizes L_k.

2.2 Generative process for the TSWE model

The formal definition of the generative process of TSWE model is as follows:

For each of sentiment-topic pair (l,z)

generate the word distribution of the sentiment-topic pair $\varphi_{l,k}$~ $Dir(\beta)$

For each document d

draw a multinomial distribution π_{d} ~$Dir(\gamma)$

For each sentiment label l under document d

draw a multinomial distribution $\theta_{d,l}$ ~$Dir(\alpha)$

For each word w_i in document d

-draw a sentiment label $l_i \sim Mul(\pi_d)$

-draw a topic z_i ~$Mul(\theta_{d,l})$

-draw a binary indicator variable s_i ~$Ber(\lambda)$

-draw a word $w_i \sim (1 - s_i)Mul(\varphi_{z_i}) + s_i MulT(\nu_{z_i}\omega^T)$

2.3 Gibbs sampling for TSWE model

In this section, we introduce the Gibbs sampling algorithm [15] for the TSWE model.The detailed derivation process on Gibbs Sampling for topic models can refer the literature [16].

The Posterior probability can be obtained from the joint probability as follows:

$$P(z_{\mathrm{i}} = k, l_i = l | w, z^{\neg i}, l^{\neg i}, \alpha, \beta, \gamma, \lambda, \nu, \omega) \propto$$
$$\left((1 - \lambda \cdot \frac{N_{l,k,w_i}^{\neg i} + \beta}{N_{l,k}^{\neg i} + V\beta} + \lambda \cdot MulT(w_i | \nu_k \omega^T) \right) \cdot \frac{N_{d,l,k}^{\neg i} + \alpha}{N_{d,l}^{\neg i} + T\alpha} \cdot \frac{N_{d,l}^{\neg i} + \gamma}{N_{d}^{\neg i} + L\gamma} \tag{3}$$

Samples derived from the Markov chain are then used to estimate π, θ and φ as depicted in equation (4), (5), (6).

$$\pi_{d,l} = \frac{N_{d,l} + \gamma}{N_d + L\gamma} \tag{4}$$

$$\theta_{d,l,k} = \frac{N_{d,l,k} + \alpha}{N_{d,l} + T\alpha} \tag{5}$$

$$\varphi_{l,k,i} = (1\text{-}\lambda) \cdot \frac{N_{l,k,i} + \beta}{N_{l,k} + V\beta} + \lambda \mathrm{MulT}(\mathrm{w}_i | \nu_k \omega^T) \tag{6}$$

3 Experiment

In this section, we explore the performance of TSWE model on document-level sentiment classification and topic extraction evaluations on different kinds of datasets for English and Chinese.

3.1 Experimental setup

3.1.1 Training word embeddings

We train 300 dimensional word embeddings on two corpus by using the Google word2vec toolkit [17]: Chinese Wikipedia[1] and English Wikipedia[2].

3.1.2 Experimental datasets

We perform experiments on two kinds of sentiment mining datasets, Chinese and English. Chinese datasets consists of three categories of product reviews datasets[3] including book, hotel, and computer, with 1000 positive and 1000 negative examples for each domain. English corpora is the polarity dataset version 2.0[4] which is introduced by Pang and Lee in 2004, consisting of 1000 positive and 1000 negative movie reviews, which we call MR04 dataset.

Preprocessing: We remove the repetitive comments and stop words, the words that word frequencies are less than 2 or larger than 15 and the words that are not found in Google embeddings representations trained from Chinese Wikipedia corpus and English Wikipedia corpus. In addition, we perform word segment for Chinese datasets

3.2 Parameter Setting

We set the symmetric prior hyper-parameter β=0.01 in our TSWE model. The symmetric hyper-parameter γ is set to $\frac{0.05 \cdot A}{L}$, where A is the average document length and L is total number of sentiment labels, as noted by [3]. The α is set to the standard setting $\frac{50}{K}$.

3.3 Experimental Results and Analysis

In this section, we present and discuss the experimental results of both document-level sentiment classification and topic extraction.

3.3.1 Sentiment classification evaluation

We use the common metrics to evaluate classification performance: Accuracy. Table 1 presents classification accuracy results obtained by TSWE on the computer data set with the number of topics K set to either 1 or 20. By varying λ, as shown in Table 1, the TSWE model obtains its best result at λ=0.1, where the λ is set 0.1 to 0.5 is better

[1] http://download.wikipedia.com/zhwiki/latest/zhwiki-latest-pages-articles.xml.bz2
[2] http://nlp.stanford.edu/data/WestburyLab.wikicorp.201004.txt.bz2
[3] http://www.datatang.com/data/11937
[4] www.cs.cornell.edu/people/pabo/movie-review-data/

than λ=0.0 on computer data sets. That shows the word embeddings is effective in capturing positive and negative sentiments. So we fix λ at 0.1, and report experimental results based on this value for the rest of this section.

Table 1. Accuracy on the computer and MR04 .

data	accuracy											
	λ	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
computer	K=1	0.765	**0.791**	**0.786**	**0.791**	**0.781**	**0.776**	0.726	0.689	0.653	0.653	0.561
	K=20	0.781	**0.797**	**0.788**	**0.782**	**0.791**	**0.786**	**0.791**	0.772	0.745	0.602	0.552

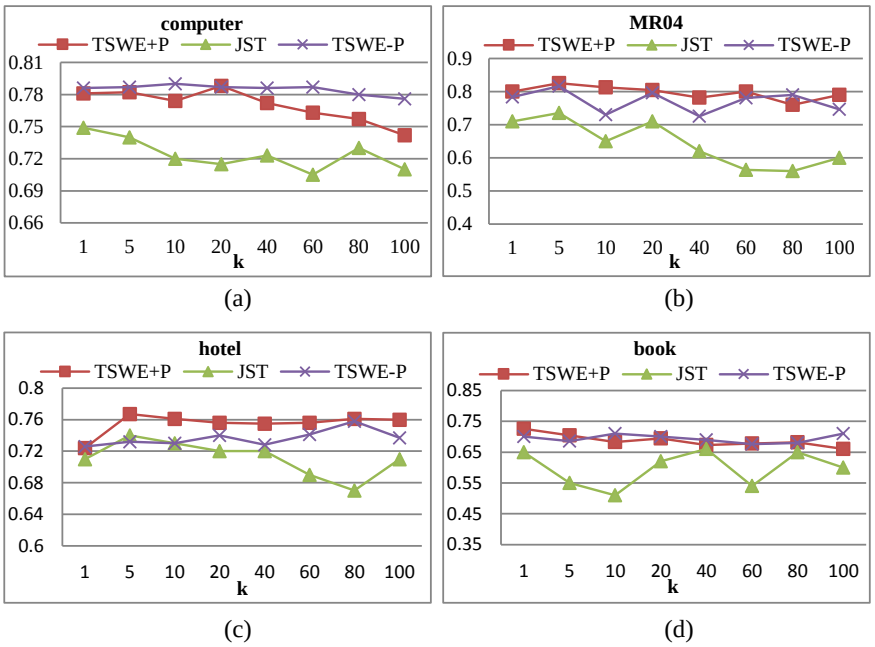

Fig. 2. Accuracy with different topic number settings on the four datasets.

With lexicons vs no lexicons:

In the experiments, we compare the classification results of introducing lexicon and no lexicon, as shown in Fig.2, TSWE+P represents the accuracy of incorporating sentiment prior, TSWE-P denotes sentiment prior is not introduced. The lexicon includes two subjectivity lexicons, the English lexicon is the MPQA[5] and the Chinese lexicon is Hownet emotional word set[6]. On most tests, the classification results of incorporating lexicon are almost similar to the the classification results of no lexicon on the same topic number. That shows the word embeddings have already captured positive and negative sentiments.

TSWE vs JST with different number of topics:

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

[6] http://www.datatang.com/datares/go.aspx?dataid=603399

Fig. 2 shows classification results produced by TSWE and the JST models on the four datasets with different numbers of topics. TSWE significantly outperforms JST in all of the datasets, particularly on the MR04 dataset where we get 20.0% improvement on accuracy at K = 80. The above results show that the word embeddings can help to extend the semantic information of words, and also can capture the sentiment information of words.

3.3.2 Topic extraction evaluation.
The other goal of evaluation task is to extract topics and evaluate the effectiveness of sentiment topic. First we need to evaluate the topic clustering performance under the corresponding sentiment polarity. We use two common metrics to evaluate the performance: perplexity and normalized mutual information (NMI) [18]. More formally, for a test set of D documents, the perplexity is:

$$perplexity = \exp\left(-\frac{\sum_{d=1}^{D}\log p(w_d)}{\sum_{d=1}^{D} N_d}\right)$$

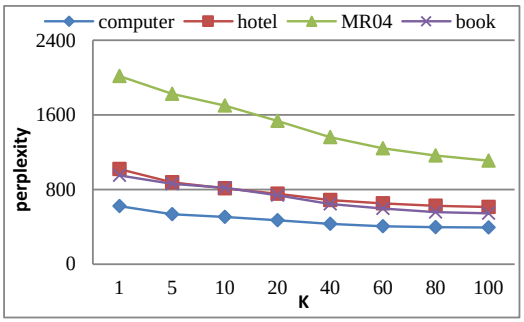

Fig. 3. Perplexity in TSWE model with different topic number settings on the four data sets

Fig. 3 shows that the perplexity on the MR04 dataset is higher than the other datasets. The reason is that the word in the corpus is more than others.

From Table 2 we can learn that the TSWE model has better NMI than JST, the NMI for TSWE model is around 0.268~0.600, and the JST obtains only around 0.10~0.420, which shows the effectiveness of the topic cluster under the sentiment with TSWE.

Table 2. NMI results in TSWE and JST on data sets book and MR04.

Data	Model	NMI							
		K=1	K=5	K=10	K=20	K=40	K=60	K=80	K=100
book	TSWE	0.392	0.353	0.309	0.338	0.293	0.302	0.315	0.268
	JST	0.260	0.083	0.070	0.195	0.270	0.062	0.24	0.168
MR04	TSWE	0.542	0.600	0.572	0.554	0.507	0.550	0.472	0.540
	JST	0.358	0.420	0.248	0.370	0.195	0.101	0.100	0.164

A topic is a multinomial distribution over words conditioned on both topics and sentiments. The most probable words for each sentiment-topic distribution could approximately reflect the meaning of the topic. Table 3 shows the selected examples of global

topics extracted from computer data set with JST and TSWE. Each row shows the top 15 words for corresponding topics. We can see that some words of TSWE such as "cooling, fan, radiator, voice, temperature, workmanship, operation" are about the computer Heat-dissipation problem, and some words such as "good, quietness, perfect, like, nice, suitable" are the emotional tendencies of the computer Heat-dissipation problem. It shows that TSWE can extract topic and sentiment simultaneously. Overall, the above analysis illustrates the effectiveness of TSWE in extracting opinionated topics under sentiment from a corpus.

Table 3. extracted topic under different sentiment labels by JST and TSWE

JST	**Pos**	漂亮/nice ; 散热/cooling; 外观/appearance; 喜欢/like; 设计/design; 配置/configuration; 比较 /very; 时尚/fashion ; 硬盘/hard disk; 噪音/noise; 内存/memory; 本本 /machine; 完美 /perfect; 钢琴/piano; 键盘/keyboard
	Neg	声音/voice;风扇/fan;温度/temperature;发热量/calorific value;散热/cooling; 硬盘/hard disk; 接受/accept; 开机/starting up; 噪音/ noise; 发热/heat; 感觉/feeling; 确实/indeed; 运行 /operation; 控制/contro; 触摸/touch
TSWE	**Pos**	散热/cooling; 风扇/fan; 不错/good; 声音/voice; 安静/quietness; 温度/temperature;完美/perfect;散热器/radiator;喜欢/like;做工/workmanship; 漂亮/nice; 运行/operation; 游戏/game; 合适/suitable; 效果/effect
	Neg	散热/cooling; 风扇/fan; 声音/ voice; 温度/ temperature; 一般/general; 不好/bad;噪音/noise;散热器/ radiator;发热量/calorific value;机器/machine; 运行/operation; 发热/heat; 游戏/game; 硬盘/hard disk; 效果/effect

4 Conclusions and Future Work

In this paper, we propose a novel unsupervised generative model (TSWE) for jointly mining sentiments, sentiment-specific topics from online reviews. To the best of our knowledge, this is the first work to model topic sentiment joint model with word embeddings. Most importantly, the experiments on real review data sets for English and Chinese show that TSWE is effective in discovering sentiments and topics simultaneously. In the future work, we will explore how to properly introduce the lexicon with HowNet lexicon to improve the performance of detecting sentiments and sentiment-specific topics.

Acknowledgements.

This research is supported by the National Nature Science Foundation of China under Grants 61472258, 61402294, National Key Technology Research and Development Program of the Ministry of Science and Technology of China (2014BAH28F05), Science and Technology Foundation of Shenzhen City under Grants JCYJ20140509172609162 and JCYJ20130329102032059.

5 References

1. Dermouche, M., Kouas, L., Velcin, J., Loudcher, S.: A Joint Model for Topic-Senti ment Modeling from Text. In: ACM/SIGAPP Symposium On Applied Computing (SAC). (2015)
2. Lin, C., He, Y.: Joint sentiment/topic model for sentiment analysis. In: Proceedings of the 18th ACM conference on Information and knowledge management, pp. 375-384. ACM, (2009)
3. Lin, C., He, Y., Everson, R., Rüger, S.: Weakly supervised joint sentiment-topic detec tion from text. Knowledge and Data Engineering, IEEE Transactions on 24, 1134-1145 (2012)
4. Pavitra, R., Kalaivaani, P.: Weakly supervised sentiment analysis using joint sentiment topic detection with bigrams. In: Electronics and Communication Systems (ICECS), 2015 2nd International Conference on, pp. 889-893. IEEE, (2015)
5. Brody, S., Elhadad, N.: An unsupervised aspect-sentiment model for online reviews. In: Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, pp. 804-812. Association for Computational Linguistics, (2010)
6. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. the Journal of machine Learning research 3, 993-1022 (2003)
7. Hofmann, T.: Probabilistic latent semantic indexing. In: Proceedings of the 22nd an nual international ACM SIGIR conference on Research and development in information retrieval, pp. 50-57. ACM, (1999)
8. Mei, Q., Ling, X., Wondra, M., Su, H., Zhai, C.: Topic sentiment mixture: modeling facets and opinions in weblogs. In: Proceedings of the 16th international conference on World Wide Web, pp. 171-180. ACM, (2007)
9. Chen, Z., Li, C., Sun, J.-T., Zhang, J., Li, C., Zhang, J., Sun, J.-T., Chen, Z.: Sentiment Topic Model with Decomposed Prior. In: SDM, pp. 767-775. (2013)
10. Nguyen, D.Q., Billingsley, R., Du, L., Johnson, M.: Improving Topic Models with La tent Feature Word Representations. Transactions of the Association for Computational Linguistics 3, 299-313 (2015)
11. Liu, Y., Liu, Z., Chua, T.-S., Sun, M.: Topical Word Embeddings. In: AAAI, pp. 2418-2424. (2015)
12. Das, R., Zaheer, M., Dyer, C.: Gaussian LDA for topic models with word embeddings. In: Proceedings of the 53nd Annual Meeting of the Association for Computational Linguistics. (2015)
13. Liu, D.C., Nocedal, J.: On the limited memory BFGS method for large scale optimiza tion. Mathematical programming 45, 503-528 (1989)
14. McCallum, A.K.: {MALLET: A Machine Learning for Language Toolkit}. (2002)
15. Walsh, B.: Markov chain monte carlo and gibbs sampling. (2004)
16. Heinrich, G.: Parameter estimation for text analysis. Technical report (2005)
17. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representa tions of words and phrases and their compositionality. In: Advances in neural information processing systems, pp. 3111-3119. (2013)
18. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to information retrieval. Cambridge university press Cambridge (2008)

Author Index