Wikipedia Based Semantic Smoothing for Twitter Sentiment Classification

Dilara Torunoğlu[1], Gürkan Telseren[1], Özgün Sağtürk[1], Murat C. Ganiz[1,2]
[1]Computer Engineering Dept.
Doğuş University
[2]VeriUs Teknoloji Ltd. Şti.
{dtorunoglu, gtelseren, osagturk, mcganiz}@dogus.edu.tr

***Abstract*— Sentiment classification is one of the important and popular application areas for text classification in which texts are labeled as positive and negative. Moreover, Naïve Bayes (NB) is one of the mostly used algorithms in this area. NB having several advantages on lower complexity and simpler training procedure, it suffers from sparsity. Smoothing can be a solution for this problem, mostly Laplace Smoothing is used; however in this paper we propose Wikipedia based semantic smoothing approach. In our study we extend semantic approach by using Wikipedia article titles that exist in training documents, categories and redirects of these articles as topic signatures. Results of the extensive experiments show that our approach improves the performance of NB and even can exceed the accuracy of SVM on Twitter Sentiment 140 dataset.**

Keywords- text classification; semantic smoothing; wikipedia; wikipedi, wiki concept, twitter corpus.

I. INTRODUCTION

Text classification is one of the important techniques to automatically organize large amounts of textual data accumulated in organizations, social media and the Internet. Text classification gaining importance with rapid increase in the usage of internet and especially social media sites such as twitter and Facebook. As a result, a tremendous amount of textual information is generated by individuals as well as the commercial entities, and organizations. One of the important and popular application areas of the text classification is the sentiment classification in which the comment texts are usually categorized as positive or negative.

Commonly used machine learning algorithms in text classification are Naïve Bayes (NB) [1], k-nearest neighbor [2], Support Vector Machines (SVM) [3]. Although SVM is one of the best performing algorithms in this domain, NB can perform better on several cases and additionally it has several advantages such as lower complexity and simpler training procedure. However, NB greatly suffers from sparsity when applied to the particularly high dimensional data as in text classification. This is especially the case when the training data consist of very short documents such as tweets and when the training set size is limited because of the cost of manual labeling processes. In order to avoid zero probability problem smoothing methods are used. Most commonly used and default smoothing technique is called Laplace Smoothing which adds one count to all terms in the vocabulary. Several other smoothing methods are proposed in order to cope with this problem in the language modeling domain such as Good Turing Smoothing [4], Jelinek-Mercer Smoothing [5], Absolute Discounting Smoothing [6] and Linear Discounting Smoothing [7] and these can be applied in NB text classification. There are also more advanced smoothing approaches called semantic smoothing which attempts to distribute probability mass to the using semantic relations [8] [9]. We base our study on the approach introduced in [8] which extracts important concepts called topic signatures from the training documents and calculates term probabilities by statistically mapping terms to topic signatures using Expectation Maximization (EM) algorithm.

In this study we significantly extend this approach by using Wikipedia article titles that exist in training documents, and furthermore categories and redirects of these articles as topic signatures. We propose a Wikipedia based semantic smoothing approach since it exploits significant amount of semantic information encoded in the relations between article titles, categories, and redirects. On extensive experiments show that our approach increases performance on accuracy than compared to NB and exceeds SVM on Twitter Sentiment 140 dataset [10].

II. BACKGROUND AND RELATED WORK

Twitter is a popular micro blogging service where users create status messages called “tweets”. Twitter becomes almost an indefinite source in text classification. In a large scale sentiment analysis study on Twitter, they have download 1,600,000 tweets and classify them as positive or negative or neutral by looking for specific words or smiley icons that expresses your opinions [10]. The data set is called Twitter Sentiment 140 dataset. We download this dataset and reduced the number of tweets in the dataset for the enrichment of Wikipedia concepts purpose. Twitter Sentiment 140 data set has 7 big categories, namely Company, Event, Location, Misc, Movie, person and product in total 1,600,000 positive, negative and neutral tweets. Therefore, we abort the number of tweets given a mostly used in 7 categories. In end, we eliminate 64,204 tweets with positive and negative class associated.

While Wikipedia has been commonly used in text classification for semantic purpose, the studies are focuses on to transform Wikipedia into a structured thesaurus. In [11], authors focus to find the Wikipedia concepts in given document. When candidate concepts have been found these are added into with their related concepts into the document where they have been found.

In paper [12], they showed how Wikipedia and the semantic knowledge it contains can be exploited for document clustering. They have created a concept-based document representation by mapping the terms and phrases within documents to their corresponding articles (or concepts) in Wikipedia.

Although there are numerous studies using English Wikipedia in semantic analysis, there are limited numbers of studies using Turkish Wikipedia (Vikipedi) for text mining. Among those authors of [9] employ a similar study as ours. They both used bag of words (BOW) model and Wikipedia enrichment on Turkish data sets where obtained from Turkish newspapers articles. They have showed that there was a slightly improvement on accuracy of NB when used Wikipedia concepts as enrichment the data compared to SVM. On the other hand, they have only used Wikipedia Articles as Wikipedia concepts which differs from our approach.

In the paper, [8] they proposed a semantic smoothing method to increase the performance of NB classification using topic signatures which are important multiword concepts such as collocations in training set. They pointed out usage of semantic smoothing with topic signatures provided slightly better accuracy results. They conduct experiments on three collections, OHSUMED, LATimes, and 20NG. They pointed out that when the size of training documents is small; the NB with semantic smoothing performs better than the NB with background smoothing and Laplacian smoothing. We are motivated by this semantic algorithm however, instead of extracting topic signatures from training data which consist of multiword phrases we use multiword Wikipedia article titles that exist in our training documents. Furthermore, we enriched the documents by adding categories and redirects.

III. APPROACH

Wikipedia[1] is a collaboratively edited, multilingual, free Internet encyclopedia supported by the non-profit Wikimedia Foundation. It has 25 million articles, 4.1 million for English only. In Wikipedia Articles there are rich information about the concept and more as synonymy and hyperlinks. Our system is similar to the system in [9]. However, we extend it to include Wikipedia categories and redirects of the articles. The overall system is presented in Figure 1.

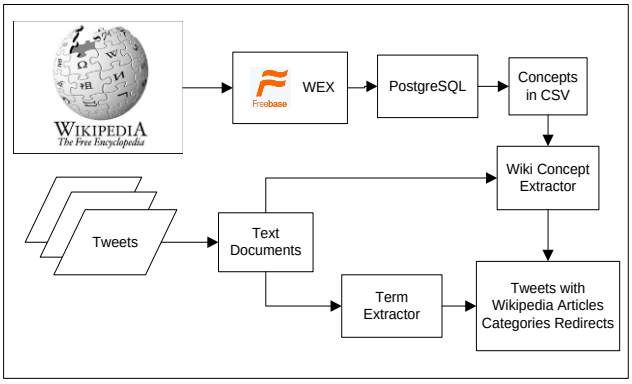

Figure 1. Design of the Wikipedia based enrichment system.

[1] http://en.wikipedia.org/wiki/Wikipedia

A. Freebase Wikipedia Extractor

The Freebase Wikipedia Extraction (WEX)[2] is a processed dump of the English language Wikipedia. Each article is transferred to readable XML format common relational features like templates, categories, article sections, and redirects are extracted in tabular form. Freebase WEX is provided as a set of database tables in TSV format for PostgreSQL[3], along with tables providing mappings between Wikipedia articles and Freebase topics, and corresponding Freebase Types.

Wikipedia dump was retrieved in August 6th, 2012 and in the PostgreSQL[3] there were 6,108,629 Wikipedia article titles, 5,587,540 Wikipedia redirects and 17,356,454 Wikipedia categories. Each of articles describes a topic which we call concepts.

We have used WEX to obtain Wikipedia tables and added them to PostgreSQL.

B. Term Extractor

Term Extractor creates an array vector that includes words term frequencies that occur in given text. These terms are added to their term-frequency vector. In Term Extractor, each text document is represented as term-frequency vector.

C. Wiki Concept Extractor

Obtained from Wikipedia dumps we have all the information on Wikipedia Articles, using PostgreSQL[3] database. Wiki Concept Extractor searches for Wikipedia article titles, categories and redirects. These concepts could be one, two or three word phrases. We have limited this search starting with two word phrases as occurrence of one word phrase would not add a semantic knowledge as it is already included in tweets. Then, given concepts are searched of occurrence in given tweets. If all the separated words of wiki concepts occur in given tweets, they are added to term frequency as if they are a unique attribute entity. By this way we exploit semantic relationships between terms in the tweets. To give an example "The White House" is the official residence and principal workplace of the President of the United States, is a Wikipedia article title. After preprocessing we obtain, "White" and "House", we check this two separated word occurrences in given tweet. Without this approach in let's say the tweet is, "My dream is to see the white house" all these words will be represented as separate terms in a vector space (Bag of Words approach) and their semantic relationship will be disregarded. On the other hand with our approach Wiki Concept Extractor will add semantic knowledge of the multiword phrase of "White house" which is associated with single words using EM algorithm in the further steps.

D. Wikipedia Articles, Categories and Redirects

As in [17], only multiword Wikipedia article titles are added to given corpus. In order to further enrich the data and obtain higher performance results Wikipedia categories and

[2] Google, Freebase Wikipedia Extraction (WEX), http://download.freebase.com/wex/, <08> <06>, <2012>

[3] http://www.postgresql.org/

redirects can be added. However, in tweet datasets it is difficult to match Wikipedia titles in these very short and noisy documents. Consequently, we only used Wiki Concept Extractor for checking the occurrence of Wikipedia article titles in given tweet. If the given wiki title is seen, then its categories are added to the document. It is important to note here that words in these categories may not exist in the original tweet document. As a result we actually extend the tweet documents and generally making them longer than 140 characters. Redirects are added similarly.

To give an example for Wikipedia article title "The White House" has 16 categories included "Houses completed in 1800", "Buildings of the United States government in Washington, D.C" and etc. With this approach we add more semantic knowledge about the "The White House" itself. Wikipedia redirects are the correct form of articles. If user writes a misspelled article, it redirects the correct article. To give an example, "Accessible Computing" is not a Wikipedia article but Wikipedia redirects user to "Computer accessibility". With this way we eliminate the misspelled word and add the correct one instead.

E. Algorithms and Formulation

In the semantic smoothing algorithm we use Wikipedia article titles, categories and redirects as topic signatures. A wiki concept is a unique meaning in a specific domain.

We use the following classifiers in our study: multinomial Naïve Bayes with Laplace Smoothing (NB) and SVM with linear kernel. Our method is called multinomial NB Wiki Semantic Smoothing (NBWS).

The semantic smoothing approach statistically maps topic signatures in all training documents of a class into single-word features using Eq. 1[8].

$$p_s(w|c_i) = (1-\delta)p_b(w|c_i) + \delta \sum_k p(w|t_k)\, p(t_k|c_i) \quad (1)$$

In Eq. 1, $p_s(w|c_i)$ is unigram class model with semantic smoothing and t_k is for *k*-th topic signature and $p(t_k|c_i)$ stands for distribution of topic signatures in training documents of a given class. Also it can be estimated by maximum likelihood estimate. δ is the coefficient for to control the influence of semantic mapping component in mixture model. If this coefficient is set to zero it turns to simple language model. In here the problem is how to compute $p(w|t_k)$. For each topic signature t_k, a set of documents (D_k) containing the signature can be obtained. Additionally, document set D_k can be used to approximate the semantic mapping from t_k to single-word features in the vocabulary. It would be unrealistic that assuming that all words appearing in $D_k$would be include in that topic signature t_k. For this reason we cannot just apply maximum likelihood estimate some words would address topics corresponding to other topic signatures while some are background words of the whole collection. To remove the noise a mixture language model was employed:

$$p(w|D_k) = (1-\alpha)p(w|t_k) + \alpha p(w|D) \quad (2)$$

For to use background collection model for generating words, α is used in Eq. 2 which is the coefficient for to use background model. The log likelihood of generating the document set D_k is:

$$\log p(D_k) = \sum_w c(w, D_k) \log p(w|D_k) \quad (3)$$

Where $c(w, D_k)$ is document frequency of term *w* in D_k, i.e., the occurrence count of *w* and t_k in all dataset. The parameters $p(w|t_k)$ can be estimated by EM algorithm [13] with following; The Expectation step we set initial values,

$$\acute{p}^{(n)}(w) = \frac{(1-\alpha)p^{(n)}(w|t_k)}{(1-\alpha)p^{(n)}(w|t_k)+\alpha p(w|C)} \quad (4)$$

For Maximization step, this process continued till it converges.

$$p^{(n+1)}(w|t_k) = \frac{c(w,D_k)\acute{p}^{(n)}(w)}{\sum_i c(w_i,D_k)\acute{p}^{(n)}(w)} \quad (5)$$

As expected, maximum likelihood estimator is initializes the EM algorithm. Regarding setting the background coefficient α. The larger α gets the more specific the trained parameters are.

IV. Experiment Setup

Originally, Twitter Sentiment 140 dataset [10] has a training set of 1,600,000 tweets and test set of 357 tweets. Tweets in the test set are about 7 categories such as products, and companies. Each category for example products consists of several product names. We have filtered training set by using queries in 7 categories. For example, for the category Product the search query was "40d" or "bing" and for company category we searched for "aig" or "at&t". The queries titles were given in [10]. There are 71 query titles that we searched in Twitter Sentiment 140 dataset and with neglecting neutral tweets we obtain 64,204 tweets with positive and negative class associated. 34,233 are labeled as negative tweets, 29,971 are labeled positive tweets.

We augmented this dataset using Wikipedia and created four different versions:

- Twitter enriched with Wikipedia article titles (TWA): Each multiword Wikipedia article title whose words exist in the tweet is added as an additional term.
- Twitter enriched with Wikipedia article titles and with respect to their categories (TWAC): In addition to the TWA, categories of existing article titles in the tweet are added.
- Twitter enriched with Wikipedia Articles and with respect to their redirects (TWAR): The same approach as TWAC however we add redirects instead of categories.
- Twitter enriched with Wikipedia Article titles, their categories and redirects (TWACR): This is the combination of TWAC and TWAR.

Table 1 shows the description of the datasets with respect to their attributes articles categories and redirect numbers.

TABLE I. DESCRIPTION OF THE DATASETS

DATA SET	# OF ATTRIBUTES	# OF ARTICLES	# OF CATEGORIES	# OF REDIRECTS
TWA	56661	4042	0	0
TWAC	68084	4042	15197	0
TWAR	70300	4042	0	12352
TWACR	83639	4042	25282	

As can be seen in the table 1, for TWA dataset we have 4,042 Wikipedia articles seen in tweets. This number is relatively very small given that there are 6,108,629 Wikipedia article titles. This may be due to the nature of tweet messages which are limited by 140 characters. Additionally, these messages are highly noisy mainly because of the informal use of language. For example, there are several words that are written informally like "nooooooooooooo", "loveeeeeeeeeee", "sundayssss", "sadddddddd", "xxplosive", "foooooooood", "okaaay" and so on. With this type of noisy terms Wiki concept extractor had difficulties to find and add the given articles titles. Additionally, we have not performed stemming which may affect the matching of article titles in tweets. In TWAC representation we have added 4,042 Wikipedia article titles and 15,197 distinct categories. Thus, the tweets are significantly enriched compared to TWA representation. In TWAR, we have added only 12,352 redirects of the 4,042 Wikipedia article titles. For the final dataset type TWACR we have added total number of 25,282 attributes included categories and redirects.

V. RESULTS AND DISCUSSION

We applied NBWS and NB and SVM algorithms to each of our datasets. For each data set we performed 10-fold cross-validation and report average accuracy. The results of our experiments are given in Figures 2 to 6.

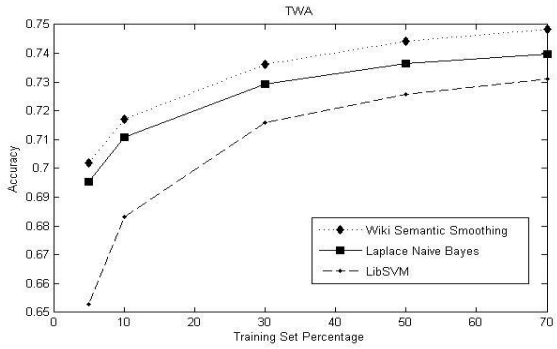

Figure 2. TWA Data Set With Algorithms.

In Figure 2, we showed the accuracies of NBWS, NB and SVM on TWA dataset, where tweets are enriched with Wikipedia articles only. As shown in the figure, performance difference between NBWS and NB increase as the training set percentage increases. However, NBWS outperforms both NB and SVM in all training set percentages.

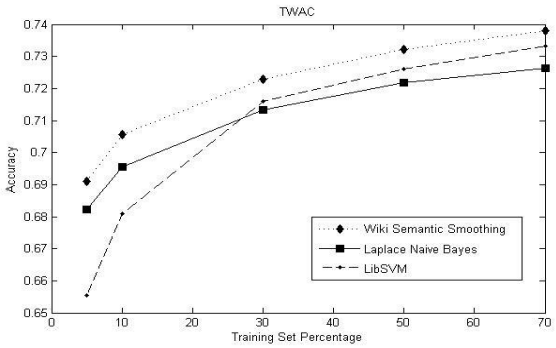

Figure 3. TWAC Data Set With Algorithms.

In Figure 3, we have a similar picture but the performance difference between NBWS and NB is more visible. Interestingly, the accuracy of SVM exceeds NB starting from 30% training set size. This is because we have more topic signatures to exploit semantic information (titles+categories) for NBWS.

In Figure 4, we have similar pattern. However, the performance gap between NBWS and NB is larger. This suggests that redirects of the articles include more semantic information than the categories of the articles.

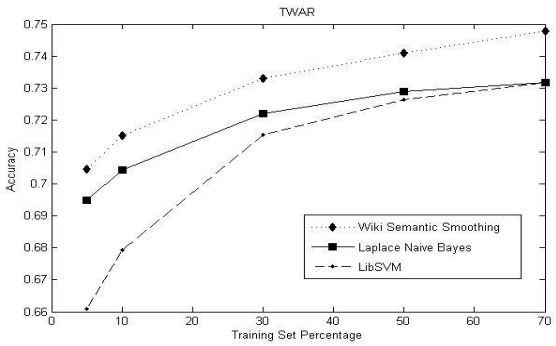

Figure 4. TWAR Data Set With Algorithms.

In Figure 5, we can see the performance of algorithms on the ultimately enriched dataset including article titles, redirects and categories in TWACR. We have even more improved performance for all the algorithms. Yet again NBWS outperforms both NB and SVM.

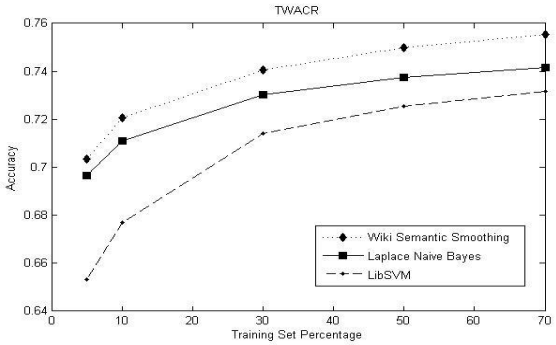

Figure 5. TWACR Data Set With Algorithms.

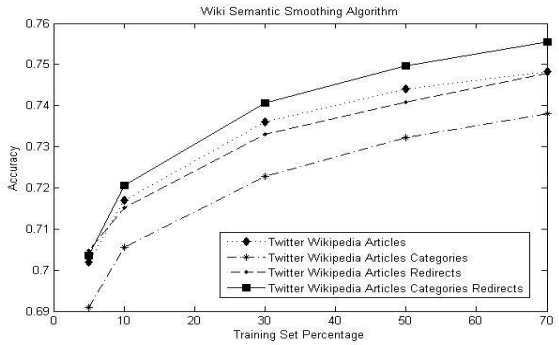

Figure 6. Wiki Semanitic Smoothing With All Data Sets.

In Figure 6, we summarize the performance of NBWS on all different versions of the twitter dataset. As we can see from here semantic smoothing benefits more from enriching the data with several different Wikipedia materials including categories and redirects. The more we add the more accuracy increased as expected. The lowest accuracy is observed on TWAC data set. Using article category knowledge for semantic enrichment decreased the accuracy. We observe that, most categories are slightly unrelated to the given article. For instance, "Barack Obama" is an article title and one of its categories is "United Church of Christ members" which is not directly related to the article and seems to be too general for the article. We speculate that this kind of categories introduce noise and therefore, decrease not only semantic meaning but also performance on accuracy.

It is important to note that these results are not the best performances that we can achieve on this data since we haven't used stemming or any kind of term weighting such as tfidf. Our main purpose is to show that we can relatively improve the performance of NB algorithm by exploiting the domain knowledge from Wikipedia for sentiment classification. In order to use domain knowledge we augmented tweets by adding multiword Wikipedia article titles as additional terms whose words exists in a tweet. This is similar to the approach in [8]. However, important concepts which are called topic signatures are acquired from Wikipedia. Next, we further enriched the tweets by adding article categories. Lastly, we add even more terms by adding redirects of the articles along with categories. When we add categories and/or redirects we may introduce terms that may not exist in the original tweet messages and therefore increase the length of the tweet message considerably. We can also argue that enrichment of tweets is of significant importance because of the harsh size limitation of 140 characters. Although one may think that it is possible to add substantial amount of noise by this approach, our results show that the algorithm performance improves by using an intelligent approach following [8]. This is because we can exploit the semantic information which is encoded into the relations between article titles, their categories and redirects by large amount of human experts.

VI. CONCLUSION AND FUTURE WORK

Sentiment classification becomes a popular application area in text classification in which comment texts are labeled positive and negative. In this domain, NB algorithm is widely used however, face a common problem; sparsity. To avoid this problem smoothing approach has been proposed. Mostly used smoothing technique is Laplace Smoothing. In this paper we propose Wikipedia Semantic smoothing approach.

Wikipedia is a very rich information resource and contains semantic relationships such as synonymy, polysemy, hyponymy, associative and categorical information and hyperlinks between articles. Using WEX we extend our Twitter Sentiment 140 dataset Wikipedia article titles, categories and redirects. After enriching the tweets we compare our proposed approach NBWS with NB and SVM on Twitter140 dataset. Results of the extensive experiments show that our approach improves the performance of NB and even can exceed the accuracy of SVM on Twitter Sentiment 140 dataset. In the future, we would like to explore this approach on Turkish Twitter datasets for sentiment classification.

REFERENCES

[1] McCallum, A., & Nigam, K. (1998, July). A comparison of event models for naive bayes text classification. In AAAI-98 workshop on learning for text categorization (Vol. 752, pp. 41-48)..

[2] Yang, Y., & Liu, X. (1999, August). A re-examination of text categorization methods. In Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval (pp. 42-49). ACM.

[3] Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. Machine learning: ECML-98, 137-142.

[4] Gale, W. A., & Sampson, G. (1995). Good‐turing frequency estimation without tears*. Journal of Quantitative Linguistics, 2(3), 217-237.

[5] Chen, S. F., & Goodman, J. (1996, June). An empirical study of smoothing techniques for language modeling. In Proceedings of the 34th annual meeting on Association for Computational Linguistics (pp. 310-318). Association for Computational Linguistics.

[6] Vilar, D., Ney, H., Juan, A., & Vidal, E. (2004). Effect of feature smoothing methods in text classification tasks. Proc. of PRIS, 4, 108-117.

[7] Manning, C. D., Raghavan, P., & Schütze, H. (2008). Introduction to information retrieval (Vol. 1). Cambridge: Cambridge University Press.

[8] Zhou, X., Zhang, X., & Hu, X. (2008). Semantic smoothing for Bayesian text classification with small training data. In SIAM2008) Proc. Intl. Conf. on Data Mining (pp. 289-300).

[9] Poyraz, M., Ganiz, M. C., Akyokus, S., Gorener, B., & Kilimci, Z. H. (2012, July). Exploiting Turkish Wikipedia as a semantic resource for text classification. In Innovations in Intelligent Systems and Applications (INISTA), 2012 International Symposium on (pp. 1-5). IEEE.

[10] Go, A., Bhayani, R., & Huang, L. (2009). Twitter sentiment classification using distant supervision. CS224N Project Report, Stanford, 1-12.

[11] Wang, P., Hu, J., Zeng, H. J., & Chen, Z. (2009). Using Wikipedia knowledge to improve text classification. Knowledge and Information Systems, 19(3), 265-281.

[12] Huang, A., Milne, D., Frank, E., & Witten, I. (2009). Clustering documents using a Wikipedia-based concept representation. Advances in Knowledge Discovery and Data Mining, 628-636.

[13] Baker, L. D., & McCallum, A. K. (1998, August). Distributional clustering of words for text classification. In Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval (pp. 96-103). ACM.