STM JOURNALS

Journal of Artificial Intelligence Research & Advances
ISSN: 2395-6720 (Online)
Volume 5, Issue 1
www.stmjournals.com

Efficient Classification of Noisy Text

Mita K. Dalal
Department of Information Technology, Sarvajanik College of Engineering and Technology, Surat, Gujarat, India

Abstract

Textual content comprises a significant volume of data generated online on a daily basis. The web-generated data often consists of high levels of noise due to a variety of factors. Development of efficient systems for automatic classification of noisy data is a crucial task in text mining. This paper examines a technique for classification of noisy text which is based on multiple feature selection and supervised learning. The main aim of the paper is to examine the efficiency of the text classification approach against increasing levels of word error rate, using both standard and web-based data sets. Empirical evaluation of the classification approach indicates that it is efficient and reliable in the presence of noise.

Keywords: *Text classification, lexical noise, machine learning, naïve Bayesian classification, acronym expansion, feature selection*

****Author for Correspondence*** E-mail: parikhmita@gmail.com

INTRODUCTION

In today's world, there is an increased reliance on the Internet for various tasks involving human interaction including inter-personal, social and commercial communications. This has led to an increased and on-going accumulation of textual data. Specifically, activities like blogging, emailing and instant messaging, online chatting, archiving of information in digitized textual form etc. have contributed to generation of web text. However, the text data is often noisy due to a variety of factors such as, use of acronyms or undocumented abbreviations, mistakes in spelling, errors in converting printed text to digitized form through optical character recognition, errors in automatic transcription of speech to text, etc. [1–3]. This paper aims to examine a supervised approach for classification of noisy text, with a focus on analyzing its performance in terms of classification accuracy against increasing levels of lexical noise.

Next part of the paper briefly surveys relevant literature in the field of text categorization; the following part elaborates on the method adopted in this paper to classify noisy text; presents the outcome of simulations done to classify noisy text and provides a performance analysis of the classification approach. Finally, the last part concludes with a pointer to future work in this area.

RELATED WORK

Text classification has been successfully performed in literature using various statistical and semantic feature extraction methods coupled with supervised machine learning approaches for classifier modeling [4–11].

In order to classify text, a representative feature set of words needs to be extracted from the text document corpus. Well known feature extraction methods for text categorization include TF*IDF (Term frequency * Inverse document frequency) [5, 6, 8, 12–14], LSI (Latent Semantic Indexing) [8, 15], Multi-word [8, 16, 17], Information Gain [10, 14] and Chi-Square [7, 14, 18]. In order to perform multi-class text categorization, a balanced feature set with sufficient features representing each category is necessary, to prevent bias. The impact of category skew on feature selection techniques for text classification has been reported in literature [11, 19]. Models for automatic text classification have been generated using various machine learning methods such as naive Bayesian [4, 7, 17], Support Vector Machines [18] [20, 21] and Artificial Neural Networks [22].

The causes for occurrence of noise during text extraction using optical character recognition (OCR) and speech transcription systems, and its impact on text categorization have been reported by Vinciarelli [1]. The sources of noise in unstructured web-based texts and the effect of 'feature noise' on text classification have been explored by researchers [2, 3].

MODEL FOR TEXT CLASSIFICATION

This section presents the design of the multi-step text classifier used for simulations depicted in this paper and discusses the steps involved.

As shown in Figure 1, the system consists of three main steps: (i) Pre-Processing, (ii) Multiple Feature Extraction and (iii) Classifier Training. The description of these steps is given next.

Pre-processing of Text

The main reasons for pre-processing text before classification are 'dimensionality reduction' and 'acronym processing'. Since all words in a language's vocabulary are potential candidates for feature set, dimensionality reduction is a necessary step to improve efficiency of text classification. In the experimental setup of this paper, dimensionality reduction is achieved through stopword removal [4, 6] and stemming, while acronym identification is done through rule-based pattern matching [23, 24]. These steps are elaborated next.

During stopword removal, functional and non-discriminating words (for example, 'a', 'all', 'an' etc.) occurring in the text were eliminated. Stopwords were identified using the USPTO list [25]. Stemming is done to conflate a word to its root form, thus effectively reducing the vocabulary (for instance, "operational" → "operate"). Stemming was performed using the Porter's suffix stripping algorithm [26].

Furthermore, it was observed that user generated web text frequently contains acronyms and abbreviations which are not present in the language dictionary. However, acronyms are important feature terms. In order to identify acronyms they were matched against glossaries of domain-specific terms using pattern matching rules listed in Table 1. The pattern matching done is case-insensitive in order to increase the chances of match.

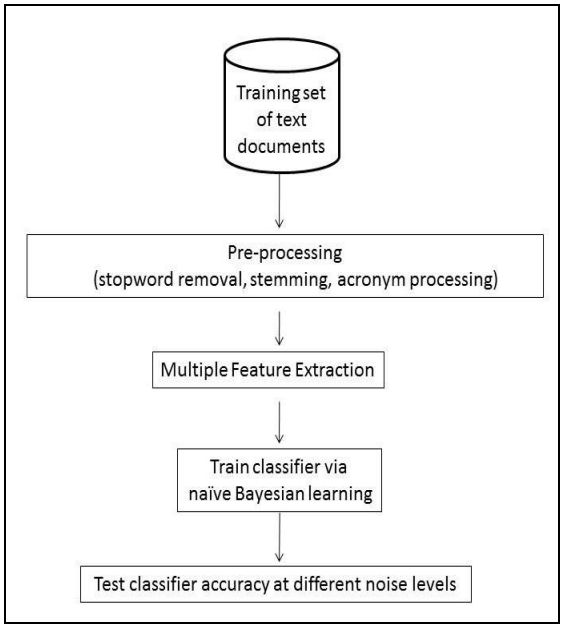

Fig. 1: *Design of Multi-step Text Classifier.*

Table 1: *Pattern Matching Rules for Expansion of Acronyms.*

Rule No.	Pattern Description	Instance
P1	Concatenate first letter of each word to form acronym	LBW (Leg before Wicket)
P2	Same as P1, but the separator between the words is a hyphen	NRR (Net-Run-Rate)
P3	Same as P1, but with one or more words replaced by digits	B6 (Boundary Sixes)
P4	Same as P1, but with stopwords ignored	BBM (Best Bowling in a Match)
P5	Same as P1, but with stopword replaced by a special character (for example, "and" replaced by "&")	C&B (Caught and Bowled)

Multiple Feature Extraction

The feature set for this text classification model is generated using multiple statistical and semantic approaches including: (i) top-ranking TF*IDF terms [5, 8, 12, 13], (ii) Multi-word features [9, 16] and (iii) frequently occurring acronyms and abbreviations identified during the pre-processing phase. TF*IDF (term frequency * inverse document frequency) is a well-known statistical measure used to determine the significance of a word in a document corpus [8, 12, 13]. TF*IDF is

based on the heuristic that a word is significant if it occurs frequently within a document, but does not occur across several documents of a category. A multi-word is a 'sequence of words' which is more indicative of domain than the individual words comprising it [6, 8]. For example, the multi-word "Indian dribble" is feature term indicative of the game of hockey.

The extracted feature set is further used to represent text documents using the multivariate Bernoulli Model [4, 6, 7]. Using the multivariate Bernoulli Model, each text document to be classified is represented as a binary vector of fixed length, wherein the occurrence of each feature term is codified as '1' if present and '0' if absent.

Classifier Training

To classify text documents, a naive Bayesian classifier [4, 6, 7] is trained. The WEKA machine learning toolkit is used for this purpose [27, 28]. The naive Bayesian model is chosen for classification in the simulations described in this paper, since it is a probability based, fault-tolerant and efficient model which can successfully be applied to the text classification task [4, 6, 7, 29].

The text datasets in the training corpus are depicted in Table 2. As shown in Table 2, the first dataset comprises of documents belonging to four categories from the well-known 20 Newsgroups dataset [4]; while the second dataset comprises of four categories of blog posts obtained by crawling the web using an automatic web crawler, with 1000 documents per category. The classification model is trained and evaluated on both datasets. The details of simulations, results and discussion are described in the next section.

EMPIRICAL EVALUATION AND RESULTS

In this section, the performance of the classification model is evaluated in the presence of lexical noise. The first part describes simulation of noise in text documents for experimentation, while the second part examines the performance of the classifier in terms of accuracy against increasing noise levels over the datasets depicted in Table 2.

Table 2: *Dataset Description.*

Sr. No.	Dataset	Categories
1.	20Newsgroups	comp.sys.ibm.pc.hardware
		rec.sport.hockey
		sci.space
		talk.politics.guns
2.	Web dataset	sports
		travel
		electronics
		finance

Simulation of Noise in Text

Web-based text frequently contains lexical noise which occurs due to errors in spelling, use of undocumented abbreviations and slang terms [3]. In order keep the study of noise in this paper realistic, typographical errors that imitate the natural mistakes made by bloggers are programmatically simulated. It has been observed that user-generated typographical errors are caused due to insertion, deletion, substitution, transposition or repetition of characters [2, 30]. For example, Figure 2 shows the original and noisy versions of a paragraph extracted from a document of the "sci.space" category of the 20Newsgroups dataset. For illustrative purposes, in Figure 1 all words altered due to noise are underlined, while the 'feature words' which got altered are both underlined and italicized.

Original text

Luna has an inconvenient gravity field. It's likely too low to prevent calcium loss, muscle atrophy, and long term genetic drift. Luna has a modest vacuum and raw solar exposure two weeks a month, but orbital sites can have better vacuum and continuous solar exposure. Luna offers a source of light element rocks that can serve as raw materials, heat sink, and shielding. The asteroids and comets offer sources of both light and heavy elements, and volatile compounds, and many are in less steep gravity wells so that less delta-v is required to reach them.

Noisy text (WER = 30 %)

Luan has an incnovenient gravity field. It's likely too loq to prevent calcium loss, muscle atreophy, and long term genetic drift. Luna has a moest vacuum anfd raw *dolar* exposure tow weeks a month, bbut orbital sitse can hve better vacuum anbd continuosu *sollar* exposure. Luna offers a sourcee of ight element rocsk that cann serve as raw materials, heat ssink, and shieldibg. Tje *asteroid* and comets offer ources of both light and heavy elemrnts, and vlatile compounds, and mnay are in less steep *graavity* wrlls so that less *dwlta-v* is required to reach them.

Fig. 2: *Paragraph with Simulated Lexical Noise.*

The level of noise in text due to typographical errors is measured using the word error rate (WER), as specified in Eq. (1) [30]:

$$WER = (I + D + S + T + R) / N \quad (1)$$

In Eq. (1), the variables *'I'*, *'D'*, *'S'*, *'T'* and *'R'* represent the number of words altered due to insertion, deletion, substitution, transposition and repetition of characters respectively, while *'N'* is the total number of words in the text document. The noise sensitivity of the proposed classification model is tested by programmatically varying the word error rate within a text document from 10 to 80%. The results of these simulations are given next.

Performance of Classification Model in Presence of Noise

In this section, the performance of the classification approach described earlier is measured. The performance is plotted as classification accuracy (F-score) vs. WER, averaged over 3-fold cross-validation. The simulations are repeated over both datasets of Table 2, with word error rate varying from 10 to 80%. Figures 3 and 4 depict these plots over the 20Newsgroups dataset and the web-crawled dataset respectively.

It can be observed from these plots that the text classification system retains an average classification accuracy of over 82% even with noise levels as high as 40% over both datasets. As shown in Figure 3, the accuracy of the classifier falls below 70% over the 20Newsgroups dataset only after noise level exceeds 60%. Similarly, it can be observed from Figure 4 that the accuracy of text classification falls below 70% over the web dataset only after the lexical noise levels reach a word error rate of 58%. Thus, the model displays very good resilience to lexical noise.

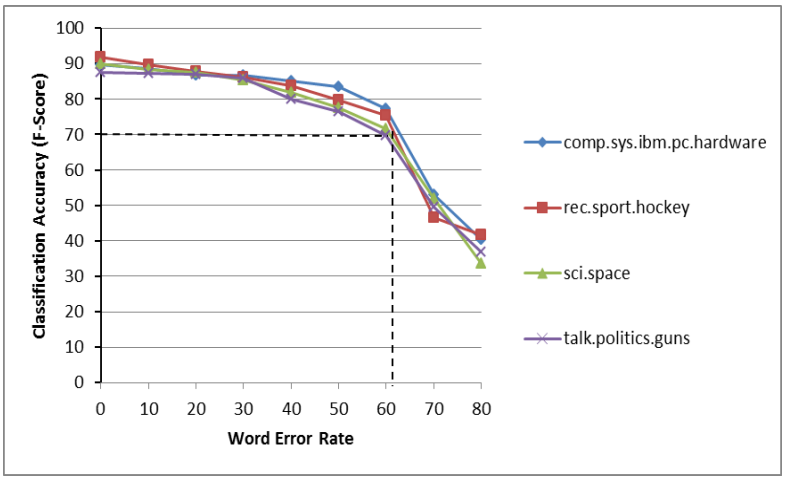

Fig. 3: *Effect of Noise on Classification Accuracy (20Ng Dataset).*

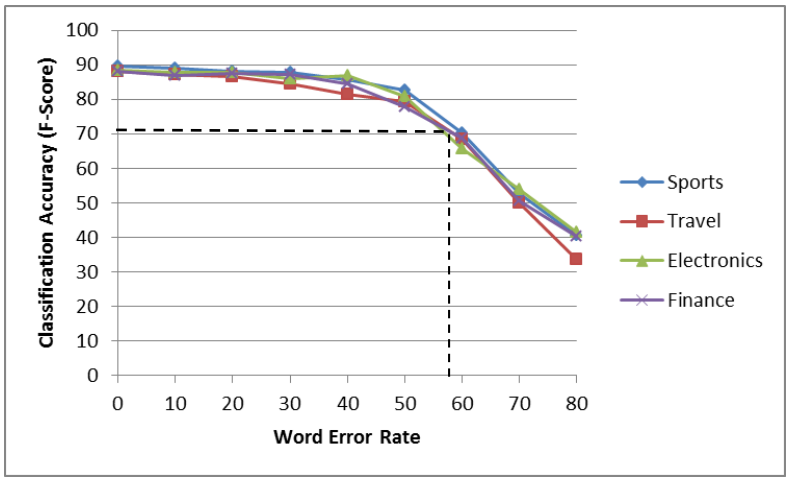

Fig. 4: *Effect of Noise on Classification Accuracy (Web Dataset).*

Upon analysis, it is found that this high tolerance to noise levels is due to two main reasons. The first reason is that all words occurring in a text are not useful in classification. Only the feature terms identified during the training phase are useful in classifying a new document. For instance, even when 40% of the words of the test set are corrupted (i.e. WER=40%), the percentage of actual feature terms corrupted is relatively less. The second important reason for higher noise tolerance is the use of the Multivariate Bernoulli model for document representation wherein the presence/absence of a feature term is represented as 1/0. So, unless all occurrences of a particular feature term are corrupted, it is not lost during document representation. To illustrate this, consider the Figure 2. It can be observed from this figure that the feature words "luna", "solar", "gravity" and "delta-v" of the original text were corrupted, while feature words "orbital", "vacuum" and "comets" were not altered. However, the features "luna", and "gravity" were not lost due to recurring occurrences of these words in the text. Thus effectively, only two features "solar" and "delta-v" were lost due to noise. The retained feature terms were sufficient to classify the document successfully to the correct category "sci.space". Thus, the discussed text classification model displays high tolerance to lexical noise and is acceptable in a web-based environment.

CONCLUSION

Textual data acquired via various digital and online sources contains lexical noise. Hence, it is necessary to develop automatic text classification systems that are resilient to noise. The multi-step classification method discussed in this paper which comprises of pre-processing for dimensionality reduction and acronym expansion, multiple feature extraction and naive Bayesian classification is empirically proven to have a classification accuracy of up to 92% in the absence of noise, and further displays a high resilience of 82% accuracy with lexical noise levels as high as 40%, which is satisfactory. In future, the author proposes generalizing the model and experiments presented in this paper to work with multi-lingual text data due to their increasing prevalence in blogs and other online data sources.

REFERENCES

1. Vinciarelli A. Noisy Text Categorization. *IEEE Trans Pattern Anal Mach Intell*. Dec 2005; 27(12): 1882–1895p.
2. Agarwal S, Godbole S, Punjani D, *et al*. How Much Noise is too Much: A Study in Automatic Text Classification. *Proceedings of the 7th IEEE International Conference on Data Mining*, Omaha. Oct 2007; 3–12p.
3. Dey L, Mirajul Haque SK. Studying the Effects of Noisy Text on Text Mining Applications. *Proceedings of the 3rd workshop on Analytics for Noisy Unstructured Text Data*, Barcelona. Jul 2009; 107–114p.
4. Kim S, Han K, Rim H, *et al*. Some Effective Techniques for Naive Bayes Text Classification. *IEEE Trans Knowl Data Eng*. Nov 2006; 18(11): 1457–1466p.
5. Lee K, *et al*. Twitter Trending Topic Classification. *Proceedings of the 11th IEEE International Conference on Data Mining Workshops*, Vancouver. 2011; 251–258p.
6. Dalal MK, Zaveri MA. Automatic Classification of Unstructured Blog Text. *Journal of Intelligent Learning Systems and Applications (JILSA)*. May 2013; 5(2): 108–114p.
7. Meena MJ, Chandran KR. Naive Bayes Text Classification with Positive Features Selected by Statistical Method. *Proceedings of the IEEE International Conference on Advanced Computing*, Chennai. 2009; 28–33p.
8. Zhang W, Yoshida T, Tang X. TF-IDF, LSI and Multi-Word in Information Retrieval and Text Categorization. *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, Singapore. Oct 2008; 108–113p.
9. Zhang W, Yoshida T, Tang X. Text Classification Using Multi-Word Features. *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, Montreal. Oct 2007; 3519–3524p.
10. Fu Z, Chen C, Gong Y, *et al*. A Comparison Study: Web Pages Categorization with Bayesian Classifiers. *Proceedings of the 10th IEEE International Conference on High Performance Computing and Communication*, Dalian. Sep 2008; 789–794p.

11. Xu Y. A Comparative Study on Feature Selection in Unbalanced Text Classification. *Proceedings of the 4th International Symposium on Information Science and Engineering*, Shanghai. Dec 2012; 44–47p.
12. Jones KS. A Statistical Interpretation of Term Specificity and Its Application in Retrieval. *J Doc*. 2004; 60(5): 493–502p.
13. Jones KS. IDF Term Weighting and IR Research Lessons. *J Doc*. 2004; 60(5): 521–523p.
14. Yang Y, Pederson JO. A Comparative Study on Feature Selection in Text Categorization. *Proceedings of the 14th International Conference on Machine Learning*, Nashville. Jul 1997; 412–420p.
15. Deerwester S, Dumais ST, Furnas GW, *et al*. Indexing by Latent Semantic Analysis. *J Am Soc Inf Sci*. 1990; 41(6): 391–407p.
16. Church KW, Hanks P. Word Association Norms, Mutual Information and Lexicography. *Comput Linguist*. 1990; 16(1): 22–29p.
17. Dalal MK, Zaveri MA. Automatic Text Classification of Sports Blog Data. *Proceedings of the IEEE International Conference on Computing, Communications and Applications*, Hongkong. Jan 2012; 219–222p.
18. Zhang B, Xu M, Wu M. Research on Web Filtering Technology Based on the Dual Feature Selection. *Proceedings of the 3rd IEEE International Conference on Network Infrastructure and Digital Content*, Beijing. Sep 2012; 675–679p.
19. Simeon M, Hilderman R. An Empirical Study of Category Skew on Feature Selection for Text Categorization. *Proceedings of the 22nd Canadian Conference on Artificial Intelligence*, Springer LNCS 5549, Kelowna, Canada. May 2009; 249–252p.
20. Ben-Hur A, Weston J. A User's Guide to Support Vector Machines. *Data Mining Techniques for the Life Sciences*. Ch. 13. Springer; 2009; 223–239p.
21. Platt JC. Fast Training of Support Vector Machines Using Sequential Minimal Optimization. *Advances in Kernel Methods*. Cambridge, USA: MIT Press; 1999; 185–208p.
22. Wang Z, He Y, Jiang M. A Comparison among Three Neural Networks for Text Classification. *Proceedings of the 8th IEEE International Conference on Signal Processing*, Beijing. Nov 2006; 1883–1886p.
23. Ratinov L, Gudes E. Abbreviation Expansion in Schema Matching and Web Integration. *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*, Beijing. Sep 2004; 485–489p.
24. Taghva K, Gilbreth J. Recognizing Acronyms and Their Definitions. *Int J Doc Anal Recognit*. May 1999; 1(4): 191–198p.
25. USPTO Patent Full Text Database. [Online]. http://patft.uspto.gov/netahtml/PTO/help/stopword.htm
26. Porter MF. An Algorithm for Suffix Stripping. *Program*. 1980; 14(3): 130–137p.
27. Hall M, *et al*. The WEKA Data Mining Software: An Update. *ACM SIGKDD Explor*. 2009; 11(1): 10–18p.
28. Witten IH, Frank E, Hall MA, *et al. The WEKA Workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques"*. 4th Edn. Morgan Kaufmann; 2016.
29. Ng AY, Jordan MI. On Discriminative vs. Generative Classifiers: A Comparison of Logistic Regression and Naive Bayes. *Proceedings of the 14th International Conference on Neural Information Processing Systems*, Vancouver. Dec 2001; 14: 841–848p.
30. Damerau FJ. A Technique for Computer Detection and Correction of Spelling Errors. *Commun ACM*. 1964; 7(3): 171–176p.

Cite this Article

Dalal Mita K. Efficient Classification of Noisy Text. *Journal of Artificial Intelligence Research & Advances*. 2018; 5(1): 56–61p.