Significance of Low Frequent Words in Patent Classification

Akmal Saeed Khattak
Natural Language Processing
Department of Computer Science
University of Leipzig
Leipzig, Germany
akhattak@ informatik.uni-leipzig.de

Gerhard Heyer
Natural Language Processing
Department of Computer Science
University of Leipzig
Leipzig, Germany
heyer@ informatik.uni-leipzig.de

***Abstract*—Low frequent terms are often considered noise but in case of patent documents it might refer to technical terms. This paper shows the significance of low frequent terms in patent classification. Our experiments show that low frequent terms cannot be ignored in patents as it give better performance in terms of f-measure and accuracy than high frequent terms. Experiments are shown to prove that set of low frequent terms outperforms set of high terms in classifying patent documents.**

Keywords - patent classification; text classification; taxonomy; International Patent Classification (IPC)

I. INTRODUCTION

The process of assignment of one or more predefined classes to text documents automatically is called text classification or categorization. There are many applications of text classification like organizing web pages into hierarchical categories, indexing journal articles by subject categories (e.g., the Library of Congress, MEDLINE, etc.), responding to Census Bureau occupations, filtering email messages, tracking news events and filtering by topics, archiving patents using International Patent Classification (IPC). Patent Classification or Categorization is one of the application area of text classification. Text classification approaches for patent classification problems have to manage simultaneously very large size of hierarchy, large documents, huge feature set and multi-labeled documents [1]. IPC is a standard taxonomy developed and maintained by World Intellectual Property Organization (WIPO). The IPC consists of about 80,000 categories that cover the whole range of industrial technologies [1]. There are 8 sections at the highest level of the hierarchy, then 128 classes, 648 subclasses, about 7200 main groups, and about 72000 subgroups at lower levels [1]. The top four levels from the 80000 classes are mostly used in automatic patent classification systems [1]. The IPC is a complex hierarchical system, with layers of increasing detail. For example, Section: G Physics, Class: G02 Optics, Subclass: G02C Spectacles, sunglasses or goggles ..., Main group: G02C5 Construction of non-optical parts.

Patent classification is a kind of knowledge management where documents are assigned predefined categories. Patent collections consist of huge vocabulary and this large vocabulary reduces the classification performance in terms of accuracy. The reason for low accuracy of classifier is due to inclusion of noisy words that is needed to be differentiate from dominant words. We reduce the vocabulary size by considering only frequent terms that have frequency above than a threshold based on some document frequency of that those terms in the entire collection. In experiments, it was found that low frequent terms can efficiently figure out dominant terms and due to inclusion of low frequent terms the classification accuracy is increased.

The remainder of this paper is structured as follows. Section II discusses related work in the field of text classification and its application patent classification. Section III gives a methodology consisting of previous algorithms to classify patents. Section IV gives analysis and experiment results. In experiment section, we discuss results on two datasets. Finally Section V is about the key lessons learned and some direction for future work for further exploration.

II. RELATED WORK

Sebastiani [2][3] has written an excellent survey on machine learning methods for text categorization and various challenges in it. Ceci and Malerba [4] investigated the issues regarding representation of documents and also the learning process. Dumais and Chen [5] explores the use of hierarchies to classify a large collection of web content. A number of statistical classifications and machine learning techniques have been applied to text categorization, including nearest

 ISBN: 978-1-61208-139-7

neighbor classifiers [6][7], Centroid-Based Classifier [8], Naive Bayes (NB) [9], Decision Trees [10] and Support Vector Machines (SVM) [11]. These machine learning techniques can be applied to patents as patent is a text document. Larkey [13] developed a classification tool based on a k-Nearest Neighbors (k-NN) approach. Chakrabarti, Dom and Indyk [14] developed a hierarchical patent classification system using 12 subclasses organized in three levels. Krier and Zaccà [15] discussed a comprehensive set of patent classification experiments. These authors organized a comparative study of various classifiers but the detailed results are not disclosed [12]. Fall, Torcsvari, Benzineb and Karetka [12] showed that instead of using full texts, the first 300 words from the abstract, claims, and description sections gives better performance regardless of classifiers.

III. METHODOLOGY

The documents are stored in many kinds of machine readable form such as PDF, DOC, Post Script, HTML, XML. The content of documents is transformed into a compact representation. Representation of text influence the classifier in achieving better performance. Text classification consists of 3 phases: text representation, building classifier model, testing classifier (evaluation). Vector Space Model (VSM) is a common way to represent document in a vector of terms [17]. Once documents are represented as a vector of terms, terms are weighted across the document collection using weighting schemes. Table 1 shows three weighting schemes TFIDF (Term Frequency Inverse Document Frequency), BM25 (Best Match) and SMART (System for Manipulating and Retrieving Text) formulas. The formulas for these weighting schemes are given in Table 1. After the assignment of weights to terms, classifiers are build on training set and using this model data is tested from the testing set. The four classifiers used are NB, SVM, Decision Trees, KNN (for k=1 and 3). The naïve Bayesian classifier is a statistical classifier [22]. Bayes' theorem is the basis for Bayesian classification [22]. The basic idea in a Naive Bayesian Classifier is the assumption that the effect of an attribute value on a given class is independent of the other values of other attributes [22]. SVM is a state-of-the-art machine learning method developed by V.Vapnik et. al. [21] is well suited for text classification [11]. The reason that SVMs work well for text classification is the huge dimensional input space, and document vectors sparsity [11]. Decision tree does not require any knowledge [23]. Given a training data a decision tree can be induced. From decision tree rules are created about the data and using these rules documents in testing set are classified [23]. Another type of classifier is an instance based classifier called K-nearest neighbor or KNN. KNN can be applied to many fields of data mining. KNN is a supervised learning algorithm. The similarity between all documents of testing and training set is computed. For each document in testing set K nearest neighbors in training documents are considered and the class is assigned based on the majority of K nearest neighbors [24]. The last step in text classification is evaluation. Using the contingency table 2, the classifiers are evaluated using the measures shown in Table 3.

TABLE 1 Different Term Weighting Schemes

Term Weighting	Formula
$w_{ij}=tf_{ij}.\log\frac{N}{n_i}$	TFIDF [18]
$w_{ij}=\frac{(k+1).tf_{ij}}{k(1-b)+b.\frac{doc_{len}}{avg_{doclen}}+tf_{ij}}.log\frac{N-df_j+0.5}{df_j+0.5}$	BM25 [19]
$w_{ij}=\frac{(1+\log(tf_{ij}))}{(1+\log(avgtf_{ij}))}.\frac{1}{\left(0.8+0.2.\frac{doc_{len}}{avg_{doclen}}\right)}.\log\frac{N}{df_j}$	SMART [20]

TABLE 2 Contingency Table

		Predicted	
		negative	positive
Actual examples	negative	a	b
	positive	c	d

TABLE 3 Evaluation Measure

Evaluation	Formula
Accuracy [2]	$A=\frac{(a+d)}{(a+b+c+d)}$
Precision [2]	$P=\frac{d}{(b+d)}$
Recall [2]	$R=\frac{d}{(c+d)}$
F-measure [2]	$F=\frac{(2.P.R)}{(P+R)}$

IV. DATASET AND EXPERIMENTAL RESULTS

Dataset-1: The main focus of these experiments are to explore the impact of low frequent terms on patent

classification in comparison with frequent terms irrespective of the hierarchical structure of patents. Only main group label of documents are considered ignoring rest of the labels (subclass, class, section). First dataset was downloaded from http://www.freepatentsonline.com [27]. For labels of documents only main group classes are considered. Documents were in HTML form. Documents contain several sections like Title, Document Type and Number, Abstract, Inventors, Application Number, Publication Date, International Classes, Claims, Description and some others. First of all, the set of patent documents (both training and testing) are preprocessed. All HTML tags are removed and hence converted to plain text. Only text under claim section of patent documents are considered here. The plain text is then preprocessed. Preprocessing extract content word. All case words are treated as small. An algorithm for suffix stripping is applied in order to perform stemming [29]. In literature, it can be found that stemming is not useful in terms of accuracy but it is useful in reducing the dimensions of text. All words that are less than or equal to 4 characters are also removed. All stop words are removed. After preprocessing a set of unique 4351 terms (word type) is obtained. Now the preprocessed text is represented in a representation model. Experiments are performed on 1484 documents. The train / test split is 66 / 34 %. Experiments are made on 4 classifiers (naïve Bayesian, support vector machine, j48, k nearest neighbor) using four weighting schemes (tfidf, bm25, smart). Following are some threshold on terms selection to investigate the effect of terms (both low and high frequent) on patent classification:

I. that occur in more than 10 document and less than 101 documents (low frequent terms)
II. that occur in more than 100 documents and less than 201 documents (frequent terms)
III. that occur in more than 200 documents (high frequent terms)

All the experiments on this dataset was carried out using WEKA [25]. The main focus was to investigate the significance of low frequent terms in comparison with frequent terms. Low frequent terms contribute more in getting better classification accuracy than frequent terms. It can be seen from Table 4 and Figure 1. Table 4 shows that in 11 out of 15 cases f-measure of classification using low frequent terms terms give better f-measure than frequent terms. This fact can be seen in Figure. 1. NB, SMO, J48 and KNN (for K=1 and K=3) classifiers when used with TFIDF weighting scheme gives better performance in terms of f-measure when terms that occur in more than 10 and less than 101 (terms that satisfies threshold I) documents are considered than terms comes under the criteria of II (terms that occur in more than 100 and less than 201 documents) and III (terms that occur in more than 200 documents) are considered.

NB, SMO and J48 classifiers when used with BM25 weighting schemes give better results for low frequent terms (I) than for high frequent terms (both II and III). The exception where frequent terms perform better than low frequent terms is when KNN (for both K=1 and 3) with BM25 is used. Similarly NB, SMO and J48 classifiers used with SMART weighting performs better in case of low frequent terms than high frequent terms. The only exception where frequent terms perform better than low frequent terms is when KNN (for K =1 and 3) is used with SMART as shown in Table 4. Only short documents in the downloaded documents are considered to make a dataset of 1484 documents. The reason behind this is WEKA is not much scalable. It just cannot classify a dataset more than 2000 documents. It got stuck in it. A patent collection can be made of both large and short documents. Thats why the LIBSVM [26] library was used in octave [28] to classify 4238 documents consisting both long and short documents

TABLE 4 F-measure on different classifiers using TFIDF, BM25 and SMART weighting schemes

	Classifier + WS	**I**	**II**	**III**
1	NB+TFIDF	**0,3730**	0,2500	0,2110
2	SMO+TFIDF	**0,3180**	0,2540	0,2540
3	J48+TFIDF	**0,3940**	0,3120	0,2640
4	KNN-1+TFIDF	**0,2640**	0,2540	0,2420
5	KNN-3+TFIDF	**0,2280**	0,2260	0,2270
6	NB+BM25	**0,3960**	0,2850	0,2610
7	SMO+BM25	**0,4510**	0,3790	0,3840
8	J48+BM25	**0,3930**	0,2730	0,2640
9	KNN-1+BM25	0,2730	0,2730	**0,2940**
10	KNN-3+BM25	0,2030	0,2550	**0,2770**
11	NB+SMART	**0,4040**	0,2850	0,2720
12	SMO+SMART	**0,4630**	0,3930	0,3250
13	J48+SMART	**0,3750**	0,3120	0,2470
14	KNN-1+SMART	0,2010	**0,2750**	0,2550
15	KNN-3+SMART	0,1690	0,2460	**0,2610**
Number of Terms		847	110	85

 ISBN: 978-1-61208-139-7

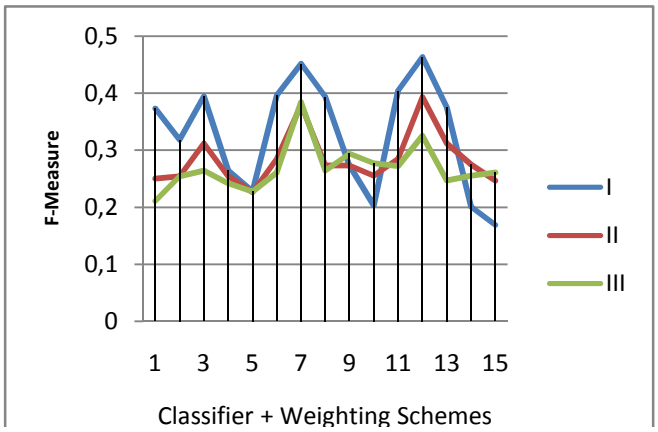

Figure 1. F-measure of different sets using classifiers and weighting schemes shown in Table 4.

Dataset-2: The other dataset is extracted from a benchmark dataset of TREC chemical patents. The total number of documents on which experimentations are made is 4238. Text were extracted for 21 main group classes. Each class have either 201 or 202 documents. Different datasets beside complete patent document consisting of various fields (title, abstract, claims, background summary, description) of patents were created. Table 5. shows word tokens and word types in each field of patent document collection on which a set of experiments were carried out. All words that are less than or equal to 4 characters are removed. All stop words are removed and stemming is performed.

LIBSVM library is used in octave to classify patent documents. 10 fold cross validation is used. The kernel type used in experiments here is linear. There are four variations of kernel in LIBSVM named as linear, polynomial, radial and sigmoid. It can be proved by experimenting that linear kernel type give better results. Two term sets or feature sets (Low Frequent Term Set LFTS and High Frequent Term Set HFTS) based on a document frequency threshold given below are created for each field and the complete patent text. The threshold criteria is as follows:

I. Terms that occur in more than 10 and less than 101 documents are considered as LFTS because it occurs between 0.24% and 2.4% documents in the entire collection.
II. Terms that occur in more than 500 and less than 1001 documents are considered as HFTS because it occurs between 12% and 24% documents in the entire collection.

The focus was to investigate the performance of low and high frequent terms and see which one gives better accuracy. It can be seen from Figure 2 and Table 6. that LFTS show better results for each field text using TFIDF, BM25 and SMART weighting scheme. Table 6. shows the performance of classifier in terms of accuracy using low and high frequent terms set on different fields of patents using TFIDF, BM25 and SMART. In Fig. 2, the blue line represents LFTS and the red line represents HFTS. Clearly it can be seen that LFTS outperforms HFTS in all cases listed in Table 6. The classification in case of title, abstract, claims, background summary, description and complete patent performs better for low frequent terms as compared to high frequent terms and gives 4.55%, 5.68%, 6.44%, 6.98%, 11.42% and 12.71% respectively better results when used with TFIDF. Similarly using LFTS with BM25 weighting scheme gives 9.49, 17.97, 4.74, 0.71, 1.35 and 3.21 percent better accuracy on all fields of title, abstract, claims, background summary, description and complete patent respectively than HFTS with BM25. Same is the case when SMART weighting scheme is used with all these fields. LFTS combined with SMART gives 9.31, 3.33, 5.19, 3.8, 3.13 and 2.48 percent better results as compared to HFTS.

TABLE 5 Word Tokens and Types in different section of patents

Field of Patent Document	Word Tokens	Word Types
Title	19717	4027
Abstract	156035	9700
Claims	761773	18488
Background Summary	2185892 (around 2.2 million)	45709
Description	5151686(around 5.2 million)	83738
All Patent Document	8283579 (around 8.3 million)	106045

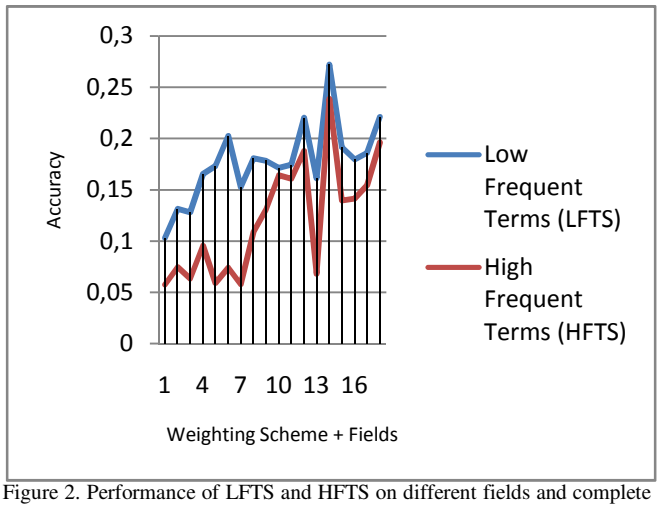

Figure 2. Performance of LFTS and HFTS on different fields and complete patents.

TABLE 6 Accuracy of Classifier on different fields (Title, Abstract, Claims, Background Summary and Description) and complete patents using TFIDF,BM25 and SMART

	Field + Weighting Scheme	Low Frequent Terms (LFTS)	High Frequent Terms (HFTS)
1	title + TFIDF	**0,1031**	0,0576
2	abs + TFIDF	**0,1314**	0,0746
3	claims + TFIDF	**0,1279**	0,0635
4	background summary + TFIDF	**0,1654**	0,0956
5	description + TFIDF	**0,1734**	0,0592
6	all + TFIDF	**0,2025**	0,0741
7	title + BM25	**0,1527**	0,0578
8	abs + BM25	**0,1808**	0,1090
9	claims + BM25	**0,1784**	0,1310
10	background summary + BM25	**0,1713**	0,1642
11	description + BM25	**0,1744**	0,1609
12	all + BM25	**0,2202**	0,1881
13	title + SMART	**0,1612**	0,0681
14	abs + SMART	**0,2721**	0,2388
15	claims + SMART	**0,1914**	0,1395
16	background summary + SMART	**0,1796**	0,1416
17	description + SMART	**0,1859**	0,1546
18	all + SMART	**0,2211**	0,1963

V. CONCLUSION AND FUTURE WORK

The main focus was to investigate the significance of low frequent terms in patent classification. Experiments above show that low frequent terms gives better performance in terms of f-measure and accuracy as compared to high frequent terms. Low frequent terms are potential discriminant terms and in patents it might refer to technical terms and might be very specific term. By selecting specific terms the classification of patents can be improved. So low frequent terms cannot be ignored as noise. In future, other threshold method like information gain and chi-square will be used for term selection and compare with the threshold based on document frequency used in this paper. The future work is to marginalize noise in patent documents to improve patent classification at different levels of IPC hierarchy specially at the main group level (higher level of details) where specific terms can improve patent classification. We also plan to investigate consider term proximity (closeness) within a document that might increase the performance of patent classification.

REFERENCES

[1] D. Tikk, G. Biró, and A. Törcsvári, "Experiment with a hierarchical text categorization method on WIPO patent collections", Applied Research in Uncertainty Modelling and Analysis, International Series in Intelligent Technologies, Volume 20, pp. 283-302, 2005.

[2] F. Sebastiani, "Machine learning in automated text categorization", in ACM Computing Surveys archive Volume 34 , Issue 1, pp. 1 – 47, 2002.

[3] F. Sebastiani, "Text Categorization", in A. Zanasi (ed.), Text Mining and its Applications to Intelligence, CRM and Knowledge Management, pp. 109-129, WIT Press, Southampton, UK, 2005.

[4] M. Ceci and D. Malerba, "Classifying web documents in a hierarchy of categories: a comprehensive study", Journal of Intelligent Information Systems Volume 28, Issue 1, pp. 37 – 78, 2007.

[5] S. Dumais and Chen, " Hierarchical classification of web content", in Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval, pp. 256– 263, New York: ACM, 2000.

[6] Y. Yand, "An evaluation of statistical approaches to text categorization", Information Retrieval, 1(1-2), 69-90, 1999.

[7] Y. Yang and X. Lin, "A re-examination of text categorization methods", In The 22nd Annual International ACM SIGIR Conference on Research and Development in the Information Retrieval, New York: ACM Press, 1999.

[8] E. Han and G. Karypis, "Centroid-based document classification analysis and experimental resultshttp://www.cs.umn.edu/wkarypis", 2000.

[9] D. Lewis, "Naive (Bayes) at forty: the independence assumption in information retrieval", In The 10th European Conference on Machine Learning, pp. 4–15, New York: Springer, (1998).

[10] D. Lewis and M. Ringuette, "Comparison of two learning algorithms for text categorization", In Proceedings of the Third Annual Symposium on Document Analysis and Information Retrieval (SDAIR'94), 1994.

[11] T. Joachims, "Text categorization with support vector machines: learning with many relevant features", In The 10th European Conference on Machine Learning, pp. 137–142, New York: Springer, 1998.

 ISBN: 978-1-61208-139-7

[12] C. J. Fall, A. Torcsvari, K. Benzineb, and G. Karetka, “Automated categorization in the international patent classification. ACM SIGIR Forum, 37(1), pp. 10–25, 2003.

[13] L. S. Larkey, “A patent search and classification system”, In Proceedings the 4th ACM conference on digital libraries, pp. 179–187, 1999.

[14] S. Chakrabarti, B. Dom, and P. Indyk, “Enhanced hypertext categorization using hyperlinks”, Proc. SIGMOD98, ACM International Conference on Management of Data, ACM Press, New York, pp. 307- 318, 1998.

[15] M. Krier and F. Zaccà, “Automatic categorization applications at the European Patent Office”, World Patent Information 24, pp. 187-196, 2002.

[16] C. J. Fall , K. Benzineb , J. Guyot , A. Törcsvári, and P. Fiévet , “Computer-Assisted Categorization of Patent Documents in the International Patent Classification”, In Proceedings of the International Chemical Information Conference, Nîmes, October 2003 (ICIC'03).

[17] G. Salton, A. Wong, and C. S. Yang, “A vector space model for information retrieval”, Communications of the ACM, 18(11), pp. 613–620, November 1975.

[18] G. Salton, and C. Buckley, “Term-weighting approaches in automatic text retrieval”, Inform. Process. Man. 24, 5, 513–523, Also reprinted in Sparck Jones and Willett [1997], pp. 323–328, 1988.

[19] S. E. Robertson, S. Walker, M. Hancock-Beaulieu, A.Gull, and M. Lau, “Okapi at TREC-3”, In Harman, D. K. (ed.) The Third Text Retrieval Conference (TREC-3) NIST, 1995.

[20] Y. H. Tseng , C. J. Lin, and Y. Lin, “Text mining techniques for patent analysis”, Information Processing and Management 43, pp. 1216–1247, 2007.

[21] V. Vapnik, “The Nature of Statistical Learning Theory”, Springer, New York, 1995.

[22] J. Han and M. Kamber, “Data Mining: Concepts and Techniques”, Elsevier, 2006.

[23] Teknomo and Kardi, “Tutorial on Decision Tree, http://people.revoledu.com/kardi/tutorial/DecisionTree, last accessed on 20.04.2011.

[24] Teknomo and Kardi, “K-Nearest Neighbors Tutorial. http://people.revoledu.com/kardi/tutorial/KNN”, last accessed on 20.04.2011.

[25] I. H. Witten and E. Frank, "Data Mining: Practical machine learning tools and techniques", 2nd Edition, Morgan Kaufmann, San Francisco, 2005.

[26] C. C. Chang and C. J. Lin, LIBSVM : a library for support vector machines, 2001. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm, last accessed on 20.04.2011.

[27] http://www.freepatentsonline.com, last accessed on 20.04.2011.

[28] http://www.octave.org, last accessed on 20.04.2011.

[29] http://snowball.tartarus.org/download.php, last accessed on 20.04.2011.

 ISBN: 978-1-61208-139-7