IR-135

Some Issues in the Automatic Classification of US Patents

Leah S. Larkey

Center for Intelligent Information Retrieval
Computer Science Department
University of Massachusetts
Amherst, Mass 01002
larkey@cs.umass.edu

Abstract

The classification of US patents poses some special problems due to the enormous size of the corpus, the size and complex hierarchical structure of the classification system, and the size and structure of patent documents. The representation of the complex structure of documents has not been a standard area of research in text categorization, but we have found it to be an important factor in our previous work on classifying patient medical records (Larkey and Croft, 1996) and in our current work on US patents.

Our classification approach is to combine the results of k-nearest-neighbor classifiers with those of Bayesian classifiers. The k-nearest-neighbor classifier allows us to represent the document structure using the query operators in the Inquery information retrieval system. The Bayesian classifiers can use the hierarchical relations among patent subclasses to select closely related negative examples to train more discriminating classifiers.

Introduction

At the Center for Intelligent Information Retrieval (CIIR) at the University of Massachusetts we are working with the US Patent and Trademark Office on a project involving the retrieval and classification of US Patent texts and patent images. This presentation focuses on the classification of patent text. This work builds upon and scales up some techniques we have used in other text categorization problems, for example, the assigning of diagnostic codes to patient medical records (Larkey and Croft, 1996) and routing and filtering (Allan et al, 1997).

The classification of US patents poses some special challenges due to three factors: the enormous size of the corpus, the size and complex hierarchical structure of the classification system, and the size and structure of patent documents. Previous work with very large numbers of documents has involved much simpler document types. For example, Fuhr's AIR/PHYS system had over a million physics articles, but they were just the titles and abstracts (Fuhr, et. al, 1991). The OHSUMED collection has around 250,000 articles from the MEDLINE database of medical journals (Hersh, et. al, 1994), and has been used in automatic indexing of around 14,000 hierarchically-related Medical Subject Headings (MeSH), (Yang, 1996) but it too contains only titles and abstracts.

DRAFT for AAAI Workshop on Learning for Text Categorization, July 27, 1998, Madison, WI.

In what follows I will describe the US patent documents and the classification system. Then I will describe some of our work on classifying US patents, emphasizing the problem of representation of patents.

US Patents

The collection

There are over 5 million US patents, consisting of 100-200 gigabytes of text. There are also more than 40 million pages of bitmap images, one image per patent page, making up 4-5 terabytes of data. We'll just be talking about the text, here, though we are also working on retrieving and classifying these images. At present we are working mostly with two years of patents, 1995 and 1996, consisting of around 220,000 documents and about 16 gigabytes in text and indices.

US Patent Documents

Patents range in size from a few kilobytes to 1.5 megabytes. They are made up of hundreds of fields, of which we represent about 50. A large number of these fields are small and not text-like, containing information like application number, patent number, dates of application, of issue, number of figures. Another large number of fields are small and contain specific pieces of text information, like the names and addresses of the authors, assignees, patent examiners, and patent attorneys. There are a few large narrative text fields, which are our primary concern:

- Title
- Abstract
- Background Summary
- Detailed Description
- Claims

As in many other real-world classification and retrieval problems, there is a severe vocabulary mismatch problem. Patents or patent applications about similar inventions can contain very different terminology. Unlike some other domains, inventors sometimes do this intentionally so their invention will seem more innovative.

The Classification System

The patent classification system consists of around 400 classes and around 135,000 subclasses. The classes and subclasses form a hierarchy, with subclasses of subclasses

of subclasses, etc. The tree goes as deep as 15 levels, but the depth varies greatly. In some cases leaf nodes are at the first level below a class, and in many places the hierarchy only goes to three or four levels. Subclasses at any level can be assigned. That is, even if a subclass is not a leaf node, and has many subclasses, which in turn have many subclasses, the parent subclass can be assigned to a patent.

A patent belongs to one class/subclass (called its *original reference*), but can also have cross references to other class/subclasses. Each patent has on average 3 cross references.

Table 1 shows the first twelve patent classes, and Table 2 shows some of the subclasses of one of those classes. The dots before the title show the hierarchical level. A title with three dots is a child of the closest preceding subclass with two dots. The classification system is dynamic. There can be up to 2000 patents in a subclass, but the patent office tries to keep it down to around 200. New inventions require the continual creation of new subclasses. Periodically, the PTO carries out a reclassification, which sometimes consists of subdividing existing classes into new subclasses, but can also involve taking a set of subclasses of a class and merging them together, and then subdividing them again in a new, more finely differentiated, manner. In either case, all the patents in the subclasses involved may or may not be assigned to the new subclasses.

CLASS	DESCRIPTION
2	Apparel
4	Baths, Closets, Sinks, and Spittoons
5	Beds
7	Compound Tools
8	Bleaching and Dyeing: Fluid Treatment and Chemical Modification of Textiles and Fibers
12	Boot and Shoe Making
14	Bridges
15	Brushing, Scrubbing, and General Cleaning
16	Miscellaneous Hardware
19	Textiles: Fiber Preparation
23	Chemistry: Physical Processes
24	Buckles, Buttons, Clasps, etc.
...	...

Table 1: First 12 patent classes

SUBCLASS	TITLE
1	Miscellaneous
2	Album Fasteners
2.5	Gun Band Type
3.1	Article Holder Attachable to Apparel or Body
4	. Chatelaine safety hooks
5	. Flower
6	.. Pin attached
7	. Napkin
8	.. Hook
9	.. Neck enclosing
10R	. Pencil
11R	.. Clasp attached
11FE	... Finger ear, belt attached pencil holder
11PP	... Pencil holder with paper clip
11CC	... Combined and convertible pencil holder

Table 2: First 15 subclasses from hierarchy for class 24

Classification tasks

The patent office is interested in automating many pieces of this process:

1. Assigning a class and subclass to a new patent application
2. Determining when reclassification needs to be performed and on what subclasses
3. Grouping or dividing existing patents into new subclasses (e.g. via clustering)
4. Reassigning cross references after a reclassification

We are currently concentrating on the first of these tasks, the assignment of documents to a patent class and subclass. The approach we are taking is to combine *k*-nearest-neighbor classification with a Bayesian or other linear classifier. These are standard classification algorithms. It is somewhat unusual to combine them, and our emphasis on document representation is innovative.

We start with *k*-nearest-neighbor because it does not require much training up front, and because it has been claimed to scale up well from small to large data sets (Yang, 1997). The Bayesian classifiers should be able to distinguish closely related subclasses, due to the selection of negative training examples from closely related subclasses. They can refine the selection made by the *k*-nearest-neighbor classifier, which tries to distinguish each subclass from all the other subclasses at once.

Categorization algorithms

k-nearest-neighbor classifier

k-nearest-neighbor classification requires a measure of similarity between patents, which in turn depends a great deal upon how documents are represented. Our *k*-nearest-neighbor classifier uses Inquery, a probabilistic information retrieval system based on Bayesian networks that uses *tf idf* weighting (Callan, Croft, and Broglio, 1994). A document to be classified is submitted to Inquery as a query. The retrieval engine returns a ranked list of documents and scores (beliefs) reflecting how good a match each retrieved document is for the test document. Inquery can take structured queries, which allows a great deal of flexibility in formulating a query from the test document, as we shall see below.

We treat Inquery's belief scores as measures of similarity, and the classes of the top k retrieved documents as the candidate classes to assign the test document. We use the belief scores to derive scores for the candidate categories by summing the scores of the documents assigned that category in the top k. Because each patent belongs to exactly one category, we then assign the top ranking category to the test document.

Bayesian Independence Classifiers

We begin with Bayesian classifiers like those we have used for medical records (Larkey & Croft, 1996) and student essays (Larkey, 1998). We train independent binary classifiers for each class/subclass using the probabilistic model described by Lewis (1992a), who derived it from a model proposed by Fuhr (1989). In our implementation, we choose a small number of features separately for each class, based on mutual information (van Rijsbergen, 1979).

A number of different research questions arise in this framework. The questions that interest us the most relate to the hierarchical structure of the class/subclass structure. Do we train classifiers for each node in the hierarchy, or just for the leaf nodes, or something in between? A central issue is what to take as the negative examples for each classifier. Do we take negative examples only from competing sibling subclasses, like Ng, Goh, and Low (1997), or sample more broadly from out-of-class examples? These issues would arise with most other classification algorithms as well, but we feel we can investigate them adequately in the context of the Bayesian model.

In addition, there are the issues of the number of features to select, and the feature quality measure.

Representation of Patent Documents

In our previous work using patient medical records (Larkey and Croft, 1996) and student essays (Larkey, 1998), we used the entire test document as a query for *k*-nearest-neighbor classification, at times using Inquery operators to differentially weight different sections of the document. For patents we do not use the entire document, or even entire sections, because many of them are too large. Instead, we reduce each test document to selected sections or portions of sections, then make a vector of the most important terms and phrases from the reduced document, and assign term weights that reflect the relative importance of the different sections the terms come from and the term frequency in those sections.

One major focus of our research is in how to make up this vector, that is, how best to represent the patents for categorization and for searching for related inventions. We are investigating the following choices in converting the document to a vector:

1. whether features should single terms only, or terms and phrases,
2. how to determine which terms (or phrases) are the best ones,
3. how many terms or phrases to include,
4. how to weight the features in the vector,
5. how to discover and represent the relative importance of different sections of the document.

The set of terms in a document is determined by first removing all occurrences of any of the 418 words on Inquery's stopword list. The remaining words were stemmed using the standard *kstem* stemmer (Krovetz, 1993). Any stem found at least twice in the patent was a candidate vector component.

Weights on terms depended upon what section of the patent it came from, and how many times it occurred in that section. A weight for the section was multiplied by the number of occurrences of the stem in the section to get a per section term weight; then the weights for that stem were summed across sections. The terms were then ranked by this weight, and a threshold (maximum number of terms) was applied to retain up to the threshold number of terms which had a weight of at least 2.

When phrases were included as features, they were chosen as follows. First, part-of-speech tags were assigned to the original document via the *jtag* tagger (Xu and Croft, 1994), and any noun phrases were flagged as potential phrases. As with the single terms, each phrase received a weight consisting of the section weight multiplied by the number of occurrences of the phrase in that section, and the weights for each phrase were summed across sections. The phrases were ranked by this weight and a threshold (possibly different from the threshold for single terms) was applied to retain up to the threshold number phrases with a weight of at least 2.

An example of a query resulting from this process for a patent on a cycle theft alarm can be seen in Figure 1. It illustrates the use of Inquery operators, #wsum, which is a weighted sum, and #1, a proximity operator meaning that terms have to occur adjacent to each other.

```
#wsum (1  11 alarm 10 switch  10 horn  10 device  6 motorcycle
6 kickstand  5 vehicle  5 button  4 lock  4 invention  4 circuit
4 battery  3 theft  3 require 3 cycle  3 close  2 weight
2 warn  2 usually  5 #1( kickstand switch)  5 #1( horn button)
5 #1( alarm device)  4 #1( lock switch)  3 #1( theft alarm)
3 #1( cycle theft alarm)  3 #1( cycle theft))
```

Figure 1: A Query formed from a Patent

Evaluation

Measures

From the point of view of the USPTO, each document has one correct class assignment. An incorrect class/subclass with the correct answer ranked second is as bad as ranking the correct answer 20th. Assigning a closely related (e.g. sibling, parent, child, etc.) class and subclass is as bad as assigning a completely unrelated subclass. A measure that reflects this absolute criterion is (microaveraged) percent correct.

We have generally found a pattern in this work in which a large number of patents are easy to categorize, and it makes very little difference what parameters, algorithms, or document representation is used. Another subset of patents are hard to classify, and we do not get those correct with any parameters or algorithms. For tuning our algorithms, it is useful to consider one condition better than another if the correct class is closer to the top of the ranking. Therefore, we use the rank of the correct class/subclass as a second measure of categorization accuracy. A rank of 1 corresponds to correct classification. We have also considered some kind of path length measure to reflect how close the proposed class/subclass is to the correct answer in the hierarchy, but we have not yet implemented this.

Test data

The USPTO has given us some test data for the placement of patent documents into subclasses. This is a relatively easy set, consisting of 469 training documents and 60 test documents from 19 different and quite distinct subclasses. We are also in the process of creating some training and test data sets from groups of closely related subclasses for finer tuning.

Some Preliminary Results

Performance on the easy set is 94% accurate using the *k*-nearest-neighbor classifier. This is, of course, an unrealistic test. When we try doing *k*-nearest-neighbor classification using the entire 1995 and 1996 complete collections to search for nearest neighbors, rather than the tiny set of 469 training documents, accuracy drops substantially. However, either corpus is useful for helping us make choices concerning the document representation, and we have used them for that purpose.

On the easy set, using only titles and abstracts, and weighting the title three times as much as the abstract, works as well as other representations using more of the document. On more difficult sets, additional portions of the document improve performance, and it appears that when sections are long, nothing is lost by using just the beginning of the section. We have not yet found, even on difficult classification tests, that the use of single terms and phrases is better than using just single terms. This somewhat surprising result is in contrast with what we have found for searching, where phrases do improve performance, at least on very short queries.

Future Work

We have a great deal more work to do on document representation, representation of the patent hierarchy, use of classifiers other than k-nearest neighbor, and using other input besides manually labeled training documents. We have so far treated the patent classification task as a standard text categorization problem, in that we have many examples of documents in and out of categories, and use the manually labeled training data to learn criteria for placing new documents automatically into these categories. However, we have several other sources of information that should help us in classifying these documents, and we are trying to use these, too. First, there is the hierarchical structure of the classification system; second, there are the names of the classes; and third, there is some additional narrative text describing criteria for assignment of documents into some classes.

We plan to use the hierarchical structure in training Bayesian or other linear classifiers whose job is to distinguish documents in one subclass from other related subclasses, rather than classifiers whose job is distinguish documents in one subclass from the rest of the patent corpus. We also plan to combine the results of *k*-nearest-neighbor classification with the results of the Bayesian classifier.

Acknowledgments

This material is based on work supported in part by the National Science Foundation, Library of Congress and Department of Commerce under cooperative agreement number EEC-9209623,and also supported in part by United States Patent and Trademark Office and Defense Advanced Research Projects Agency/ITO under ARPA order number D468, issued by ESC/AXS contract number F19628-95-C-0235.

Any opinions, findings and conclusions or recommendations expressed in this material are the author(s) and do not necessarily reflect those of the sponsor(s).

References

Allan, J., Callan, J., Croft, W. B., Ballesteros, L., Broglio, J., Xu, J., and Shu, H. INQUERY at TREC-5. In *Proceedings of TREC-5,* 1997.

Callan, J., Croft, W. B., and Broglio, J. (1994). TREC and TIPSTER Experiments with INQUERY. *Information Processing and Management,* 31(3), 327-343.

Fuhr, N. (1989). Models for retrieval with probabilistic indexing. *Information Processing and Management,* 25(1), 55-72.

Fuhr, N. , Hartmann, S., Lustig, G., Schwantner, M., Tzeras, K. (1991). AIR/X – A rule-based multistage indexing system for large subject fields. *Proceedings of the RIAO '91,*
Barcelona, Spain. pp. 606-623.

Hersh, W., Buckley, C., Leone, T. J., & Hickam, D. Ohsumed: An Interactive Retrieval Evaluation and New Large Test Collection for Research. *Proceedings of ACM SIGIR '94.* pp. 192-201.

Krovetz, R. (1993) Viewing Morphology as an Inference Process. In *Proceedings of ACM SIGIR '93*, pp. 191-203.

Larkey, L. S. (1998). Automated Essay Grading using Text Categorization Techniques. UMass CIIR Technical Report IR-121.

Larkey, L. S., and Croft, W.B. (1996). Combining classifiers in text categorization. In *Proceedings of ACM SIGIR '96* pp. 289-297.

Lewis, D. D., Schapire, R., Callan, J., and Papka, R. (1996). Training Algorithms for Linear Text Classifiers. In *Proceedings of ACM SIGIR '96,* pp. 298-306.

Lewis, D. D. (1992a). An Evaluation of Phrasal and Clustered Representations on a Text Categorization Task. In *Proceedings of ACM SIGIR '92,* pp. 37-50 .

Lewis, D. D. (1992b). *Representation and Learning in Information Retrieval.* Amherst: PhD Thesis, University of Massachusetts.

Ng, H. T., Goh, W. B., and Low, K.L. Feature Selection, Perceptron Learning, and a Usability Case Study for Text Categorization (1997). *Proceedings of ACM SIGIR '97,* pp. 67-73.

van Rijsbergen, C.J. (1979). *Information Retrieval.* London: Butterworths.

Xu, J. and Croft, W. B. (1994). The Design and Implementation of a Part of Speech Tagger for English. UMass CIIR Technical Report IR-52.

Yang, Y. (1996). An evaluation of statistical approaches to medline indexing. *Proceedings of the 1996 Annual Full Symposium of the AMIA*, pp.358-362.

Yang, Y., (1997). An Evaluation of Statistical Approaches to Text Categorization. CMU Technical Report CMU-CS-97-127.