Some Issues in the Automatic Classification of U.S. Patents

Working Notes for the AAAI-98 Workshop on Learning for Text Categorization

Leah S. Larkey

Center for Intelligent Information Retrieval
Department of Computer Science
University of Massachusetts
Amherst, Mass 01003
larkey@cs.umass.edu

Abstract

The classification of U.S. 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 received a great deal of previous attention, but we have found it to be an important factor in our work. We are exploring ways to use this structure and the hierarchical relations among patent subclasses to facilitate the classification of patents. Our approach is to derive a vector of terms and phrases from the most important parts of the patent to represent each document. We use both *k*-nearest-neighbor classifiers and 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 U.S. Patent and Trademark Office on a project involving the retrieval and classification of U.S. 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 U.S. 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.

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

U.S. Patents

There are over 5 million U.S. 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. Some of our work uses two years of patents, 1995 and 1996, consisting of around 220,000 documents and about 16 gigabytes in text and indices. Other work uses thirteen years of patents, from 1985-1997.

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, and these 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. To compound the problem, some inventors intentionally use nonstandard terminology so their invention will seem more innovative.

The Patent 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 places there is only one level of subclasses below a class, and in many places there are only three or four levels. Subclasses at any level can be assigned to patents. That is, even if a subclass has subclasses of its own, the parent subclass can be assigned to a patent.

A patent belongs to one class/subclass called its *original reference*. In addition, it can have cross references to other class/subclasses. The average patent has three cross references.

Table 1 shows a sample of patent classes. Table 2 shows some of the subclasses of one of those classes. In our preliminary work, we have been focusing on these speech-related subclasses of class 395, *Information Processing System Organization*. In Table 2, hierarchical level is indicated by indentation. Note that the subclass numbering scheme does *not* capture the hierarchical relations among 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.
…	…
395	Information Processing System Organization
396	Photography
399	Electrophotography
…	…

Table 1: A sample of patent classes

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 making new subclasses. 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 different manner. In either case, all the patents in the subclasses involved may or may not be assigned to the new subclasses.

Classification tasks

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

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

- **2.090 SPEECH SIGNAL PROCESSING**
  - **2.1 For storage or transmission**
    - …
  - **2.4 Recognition**
    - 2.41 Neural network
    - 2.42 Detect speech in noise
    - 2.43 Normalizing
    - 2.44 Speech to image
    - 2.45 Specialized equations or comparisons
      - 2.46 Correlation
      - 2.47 Distance
      - 2.48 Similarity
      - 2.49 Probability
      - 2.50 Dynamic time warping
      - 2.51 Viterbi trellis
    - 2.52 Creating patterns for matching
      - 2.53 Update patterns
      - 2.54 Clustering
    - 2.55 Voice recognition
      - 2.56 Preliminary matching
      - 2.57 Endpoint detection
      - 2.58 Subportions
      - 2.59 Specialized models
    - 2.6 Word recognition
      - 2.61 Preliminary matching
      - 2.62 Endpoint detection
      - 2.63 Subportions
      - 2.64 Specialized models
        - 2.65 Markov
        - 2.66 Natural language
  - **2.67 Synthesis**
    - …
  - **2.79 Application**
    - …

Table 2: Sample of subclasses for speech-related patents in class 395.

We are currently concentrating on the first of these tasks, the assignment a patent class and subclass to patents and other documents. The approach we are taking is to combine *k*-nearest-neighbor classification with Bayesian or other linear classifiers. These are standard classification algorithms, but 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 and Croft 1996) and student essays (Larkey 1998). We train independent classifiers for each class/subclass using the probabilistic model described by Lewis (Lewis 1992a), who derived it from a model proposed by Fuhr (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 system. Do we classify a patent based on the output of the single best classifier, or based on the best path through the subclass hierarchy, 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:

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

One example of a query made from a patent on a motorcycle theft alarm can be seen in Figure 1. It illustrates the use of two Inquery operators, #wsum, a weighted sum, and #1, a proximity operator requiring that terms 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

Such queries were constructed in the following way. The set of terms in a document was determined by first removing all occurrences of any of the 418 words on Inquery's standard 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.

The weights on features (stemmed term) 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 feature in the section to get a per section feature weight; then the weights for that feature were summed across sections. The features 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.

Some Preliminary Results

We have selected a part of the patent database for some initial experimentation consisting of all the patents in class 395, subclasses 2.09-2.89 as shown in Table 2. We have been using patents from the years 1985 through 1995 for training, and patents from the years 1996 and 1997 for testing. Although this corpus is much smaller that the full set, it is useful for helping us make choices about the document representation and classification algorithms, and we have used them for that purpose.

Concerning representation, we have settled for the present on a very small portion of each patent document. We are using a vector made up of the most frequent terms from the title, the abstract, the first twenty lines of the background summary, and the exemplary claim(s), with the title receiving three times as much weight as the rest of the text. We have not yet found that the addition of 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.

Concerning the hierarchical structure of the subclass system, we have not yet found any multilevel algorithm that performs significantly better than one which tries to choose among all the speech subclasses, but there is a great deal more work to be done.

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 sponsors.

References

Allan, J., Callan, J., Croft, W. B., Ballesteros, L., Broglio, J., Xu, J., and Shu, H. 1997. INQUERY at TREC-5. In *Proceedings of The Fifth Text REtrieval Conference (TREC-5)*, 119-132. Gaithersburg, MD.: NIST special publication 500-238.

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. In *RIAO 91*: *Intelligent Text and Image Handling*, 606-623. Barcelona, Spain.

Hersh, W., Buckley, C., Leone, T. J., & Hickam, D. 1994. Ohsumed: An Interactive Retrieval Evaluation and New Large Test Collection for Research. In *Proceedings of the Seventeenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, 192-201.

Krovetz, R. 1993. Viewing Morphology as an Inference Process. In *Proceedings of the 16th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, 191-203. Pittsburgh: ACM Press.

Larkey, L. S. 1998. Automated Essay Grading using Text Categorization Techniques. CIIR Technical Report, IR-121, Dept. of Computer Science, Univ. of Massachusetts.

Larkey, L. S., and Croft, W.B. 1996. Combining Classifiers in Text Categorization. In *Proceedings of the 19th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, 289-297.

Lewis, D. D. 1992. An Evaluation of Phrasal and Clustered Representations on a Text Categorization Task. In *Proceedings of the Fifteenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, 37-50.

Ng, H. T., Goh, W. B., and Low, K.L. 1997. Feature Selection, Perceptron Learning, and a Usability Case Study for Text Categorization. In *Proceedings of the 20th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, 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. CIIR Technical Report, IR-52, Dept. of Computer Science, Univ. of Massachusetts.

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

Yang, Y. 1997. An Evaluation of Statistical Approaches to Text Categorization. Technical Report, CMU-CS-97-127, Computer Science Department, Carnegie Mellon University.