# SVM Based Learning System for F-term Patent Classification

Yaoyong Li, Kalina Bontcheva and Hamish Cunningham
Department of Computer Science, The University of Sheffield
211 Portobello Street, Sheffield, S1 4DP, UK
{yaoyong, kalina, hamish}@dcs.shef.ac.uk

## Abstract

*This paper describes our SVM-based system and the techniques we used to adapt the approach for the specifics of the F-term patent classification subtask at NTCIR-6 Patent Retrieval Task. Our system obtained the best results according to two of the three measures used for performance evaluation. Moreover, the results from some additional experiments demonstrate that our system has benefited from the SVM adaptations which we carried out. It also benefited from using the full patent text in addition to using the F-term description as extra training material. However, our results using an SVM variant designed for hierarchical classification were much worse than those achieved with flat SVM classification. At the end of the paper we discuss the possible reasons for this, in the context of the F-term classification task.*

## 1 Introduction

Automatic processing of patent information is very useful in industry, business, and law communities, because intellectual property is crucial in knowledge based economies and the number of patent documents is huge and increasing rapidly. Machine learning algorithms have been successfully used for information retrieval and natural language processing. Patent information processing is a sub-area of automatic text processing. in which machine learning would play a key role.

Patent information processing has some unique features in comparison with general text processing. One feature is that patents can be regarded as semi-structured documents, in which different kinds of content (e.g. the purpose, method, function and effect) of each patent application are put into different sections (or subsections) with a proper title. Patents are also often associated with one or more classification schemes, in which the classes are organised in a hierarchical fashion. Moreover, there are some specific tasks in patent information processing which lead to different settings for the machine learning algorithms from the general text processing tasks (the F-term classification subtask at NTCIR-6 Patent Retrieval Task is one such example — see the discussions in Section 3.2). Therefore, in the applications of machine learning to patent information processing, we have to take into account those characteristics of patent documents in order to achieve the best performance.

This paper describes our machine learning-based participating system for the F-term patent classification subtask at NTCIR-6 Patent Retrieval Task. Section 2 briefly discusses the classification subtask. Section 3 describes our participating systems in detail, including the feature extracted from patent and the machine learning techniques. Section 4 presents our system's results on the task and other experimental results showing the benefits of several techniques in our system. Finally Section 5 gives some discussions and conclusions.

## 2 F-term classification subtask

F-term classification is one of the two subtasks of NTCIR-6 Patent Retrieval Task. For more details about the subtask see the overview paper [2] for the NTCIR-5 and the subtask overview paper in this proceedings.

Patent classification is very important for patent processing and application. The most common classification taxonomy of patent is the International Patent Classification (IPC) from the World Intellectual Property Organization. IPC is solely based on the contents of inventions. However, some patent processing or utilisation task may focus on various viewpoints of a patent, such as purpose, means, function, or effect of the invention.

To this end, the Japan Patent Office provides a two-level classification scheme for patent. The first level denoted as FI is an extension of IPC, which refers to a set of themes about patent. For example, the theme *2C088* is about "Pinball game machines (i.e., pachinko and the like)". And the theme *5J104* denotes the technical field of "Ciphering device, decoding device and privacy communication". Each theme has a collection of viewpoints for specifying possible aspects of the patent within the theme. Each viewpoint has a list of possible elements. Those viewpoints and the corresponding elements for one theme are encoded by the F-terms of the theme, which are the second level of the patent classification scheme. The viewpoints are different from one theme to another. Each particular viewpoint may consist of several elements, which are organised in a tree structure. The theme *2C088* has the viewpoint *AA* for “Machine detail”, the viewpoint *BA* for “Processing of pachinko ball”, and the viewpoint *BB* for “Card systems”. The viewpoint *AA* has the elements such as *AA01* for “Standard pachinko games (i.e., vertical pinball machines)” and *AA65* for “Special pachinko games“. Hence, the F-terms under one theme have the specific/general relations among them.

In the F-term classification, patents are first classified into themes. Given a theme a patent belongs to, the patent is further classified into the F-terms of that theme. A patent may have one or more themes and have many F-terms for each of them.

The F-term classification subtask at NTCIR-6 Patent Retrieval Task was to assign the suitable F-terms to the test patent document, given the theme(s) of the patent. It uses the 1993 – 1997 UPA Japanese patents for training and the 1998 – 1999 UPA patents for evaluation. The English translation of the abstracts of the same Japanese patent applications are also provided by the organisers, which can be used as surrogate text for the task. There are about 1200 valid themes and every theme may have several hundreds F-terms in many cases.

In the dry run, only two themes were used, namely *5J104* and *5F033*, which has 271 F-terms and 1920 training documents and 620 F-terms and 7314 training documents, respectively. We noticed that a part of relations among the F-terms of theme *5F033* was not available from the related documentation, that would make it impossible to explore the hierarchical structure of F-terms under the theme. This problem was reported to the organiser and the following assurances had been made for the formal run evaluation data, which enabled the participants to evaluate the hierarchical learning algorithm and new evaluation measures by exploring the hierarchical relations among the F-terms.

One hundred and eight themes were selected for the formal run. The numbers of the F-terms for the themes are between one and eight hundred. The number of training documents for the themes are between one and ten thousand. As expected, all F-terms and their relations are available from the issued documentation. Therefore we can explore the relations of the F-terms in classification algorithm as well as in the evaluation measure for the formal run.

# 3 Our Systems for F-term classification

## 3.1 Extracting features from patent document

The NTCIR-6 patent classification subtask used the Japanese patent documents. It also gave the participants the so-called PMGS documents (see e.g. http://www5.ipdl.ncipi.go.jp/pmgs1/pmgs1/pmgs_E) which include a brief description (several words in most cases) for each F-term and the hierarchical relations among the F-terms under each theme. Our participating systems used those two types of information released by the task organisers.

The Japanese patent document is semi-structured in the sense that it consists of many sections, each of which addresses one specific aspect of a patent application. For example, almost every patent has an abstract section containing a concise description of the patent application. Another section describes the patent in detail, which often consists of several subsections for different aspects of the patent application such as the purpose, function and implementation of the patent. A patent document also usually contains some information about the patent applicants, e.g. the name and address of the applicant and their associated company.

Our participating system was based on the patent’s content, meaning that it did not use the information about the applicant and the company, though this kind of information may possibly be useful for patent classification, as one particular applicant or company tends to apply for the same types of patents. Actually our system uses the full content of the patent documents with two exceptions. One exception is the bibliographical information and the other is the part of the text possibly containing the category codes, which had to be ignored according to the rules from the task organisers.

In detail, we first collected the titles of the sections and subsections from the training documents and then classified them into seven categories. The *abstract* and *claim* categories contain the text from the two sections, respectively. The other four categories, *technological-field*, *purpose*, *method* and *effect*, are from the corresponding sub-sections in the detailed description section. Another category *implementation* was about implementing details of the patent, such as structure of invention and implemented examples.

We also used the short description of each F-term as additional training material in the two of our four submitted runs. What we did was to treat the description text of each F-term as an extra document for training.

We then preprocessed the selected Japanese text of each document using the Japanese morphological analysis software Chasen version 2.3.3 (see http://chasen.aist-nara.ac.jp/). From the documents processed by the Chasen, we picked up as our feature terms those words whose part of speech tags were either noun (but not dependent noun, proper noun or number noun), or independent verb, or independent adjective, or unknown, as what was done in [6]. We also removed the Japanese terms appearing less than three times in the documents for training. Then we computed the $tf * idf$ feature vectors for the Japanese patent document or the description text of one F-term in the usual way (e.g. see [3]) and finally normalised the feature vectors, which were the input to the SVM learning algorithm our system used.

### 3.2 SVM based learning algorithms

Our participating systems are based on the Support Vector Machines (SVM). The SVM is a supervised learning algorithm which achieves state of the art results for many classification applications including document classification (see e.g. [3]). However, there are some differences between the conventional document classification and the NTCIR-6 F-term patent classification task. Thus we had to adapt the SVM to the specific settings of the task.

First, note that in the application of the SVM to document classification, an SVM classifier is often learned for one category, which then is used to classify one document into the category or not. In the conventional document classification, the measure of the results is based on category. In another word, for one category, it counts how many documents in evaluation set the classifier classifies correctly (or incorrectly). In contrast, in the F-term classification the result is measured for each patent document, and then a macro-averaged overall number is computed from the results of all the evaluation documents.

Secondly, as we learn one SVM classifier for one F-term by using the one vs. all others strategy, the classification problem for one F-term in many cases has an imbalanced training data in which the positive examples are outnumbered by the negative examples. The experiments in [5] showed that the SVM with uneven margins can achieve higher F-measure than the original SVM for the imbalanced training data. Hence we used the SVM with uneven margins in our systems, instead of the standard SVM.

Thirdly, as there are specific or general relations among the F-terms within one theme, it is desirable that, if one document cannot be correctly classified into one F-term, the document is classified into an F-term which is closely related to the true F-term. Hence, we would like to experiment the learning algorithm which takes into account the relations among the classes.

Finally, since a patent document contains many types of information about the patent, such as the information about the patent applicant(s), the information about the invention itself, and the typical application scenarios of the invention, we have to decide what information will be used in the F-term classification system. Moreover, as there is a short description for each F-term in the PMGS (which is the documentation about the F-terms and was provided to the participants by the subtask organisers), we want to assess if those F-term descriptions are useful in the F-term classification.

In detail, we first learn an SVM classifier for each F-term within one theme from the training documents. Then, given a patent document of the theme, we apply each of the F-term classifiers to the document and obtain a confidence score of the document belonging to the corresponding F-term as well as a classification decision if or not the document has the F-term. We can then obtain a rank of F-terms according to the confidence scores for the document. Finally several measures such as the A-Precision, R-Precision and F-measures are computed for the F-terms assignments to the document by system. Both the A-Precision and R-Precision are computed from the rank of F-terms, while the F-measures are obtained from the classification decisions of the F-term SVM classifiers for the document. In order to obtain an ordered sequence of F-terms for one document, we have to compare the confidence scores of different SVM classifiers. To make the comparison more sensible, we first normalised the output of the SVM (before thresholding) with respect to the weight vector of the SVM classifier, and then convert the normalised output into a value in between 0 and 1 via a Sigmoid function the Sigmoid function $s(x) = 1/(1 + exp(-\beta x))$ where $\beta$ was set as 2.0 in our experiments.

Based on the above considerations we obtained and submitted four runs for the formal run of the NTCIR-6 patent classification task, with the ids as GATE01, GATE02, GATE03 and GATE04. All the four runs used the normalised confidence scores for forming the rank of F-terms for one patent. They all also used the uneven margins SVM model. In the following we describe the four runs in an order of increasing complexity and highlight the differences between them.

**GATE04** The run GATE04 was the most simple one. We used the flat classification in this run, namely training one SVM classifier for each F-term by using the documents with the F-term as positive examples and all other documents in the training set as negative examples. It only used the training documents from the 1993 – 1997 UPA Japanese patent collection.

**GATE03** This run used the same flat classification scheme as the run GATE04. On the other hand, it also used the short PMGS description of one F-term as an extra positive example for training the SVM classifier for that F-term, besides the training documents from the patent collections.

**GATE02** The run GATE02 learned and applied the

SVMs in a hierarchical fashion. In another word, it use a variant of the SVM called H-SVM which was designed for hierarchical classification (see [1]). As the F-terms under one theme have the general/specific relations, they can be organised in a hierarchical fashion. In the H-SVM we first learned the SVM classifier for each of the most general F-terms, by selecting as positive examples the training documents with either the F-term itself or one F-term which is the specification of the F-term considered, and all other training document as negative examples. For one less general F-term, we learned one SVM classifier by using only those training documents which belong to its parent F-term, in which the positive examples were those documents with the F-term considered or one F-term which was the specification of the F-term.

In the application of the H-SVM for the F-term classification task, we first classified test document using each F-term classifier. Then we tried two different ways to obtain the confidence score of one document for each F-term. The first way was to use the confidence score of the F-term classifier itself. Another way was to average the confidence scores of the F-term classifier itself and all the classifiers of its ancestor F-terms. As our preliminary experiments on the training data showed that the second way obtained better results than the first way (also see the results presented in Section 4), we adopted the second way in our submitted runs. Once we obtained the confidence scores for one document and each of the F-terms, we can easily obtain an ordered sequence of F-terms and a classification decision on the F-terms.

In comparison to the flat classification, the H-SVM takes into account the relations between the class labels in both training and application. So, we can expect that, if a document cannot be classified correctly into an F-term, the H-SVM would have more tendency than the flat SVMs to classify the document into an F-term which is closely related to the true F-term.

The run GATE02 only used the training documents from the patent collections.

It is worth noting that [1] has used the H-SVM for hierarchical document classification and obtained higher F-measure than the SVM using flat classification.

**GATE01** The run GATE01 used the H-SVM, just as the run GATE02. It also used the PMGS F-term descriptions for training, besides the patent documents. It used the F-term descriptions in a different way from the GATE03. To learn an F-term classifier, it used the descriptions of the F-term itself and all its descendant F-terms — each of those descriptions was regarded as one positive training document for the training.

# 4 Results

## 4.1 Results of the four submitted runs

Table 1 presents the results of our four submitted runs, measured by A-Precision, R-Precision and F-measures, respectively. It also lists the results of one run from another participating team which had the highest A-Precision score.

First, comparing against other submitted runs (see the overview paper of the NTCIR-6 f-term classification task in this proceedings), our run GATE03 obtained the best results of the R-Precision and $F_1$, and was only slightly lower than the highest A-Precision figure of all submitted runs.

Secondly, the runs using the F-term description as additional training material obtained better results than the runs which used the same learning algorithm but did not use the F-term descriptions in training. In another word, the GATE01 and GATE03 performed better than the GATE02 and GATE04, respectively. We can conclude that the F-term descriptions were indeed helpful for the F-term patent classification.

**Table 1. The official results of our four submitted runs, together with one submitted run from another group which has the best A-Precision results. Note that one of our runs, GATE03 has the highest scores of R-Precision and $F_1$ and the second best score of A-Precision among all the submitted runs.**

| Run-ID | A-Precision | R-Precision | $F_1$ |
|---|---|---|---|
| GATE01 | 0.2376 | 0.2164 | 0.2257 |
| GATE02 | 0.2132 | 0.1860 | 0.2135 |
| GATE03 | 0.4779 | 0.4363 | 0.4125 |
| GATE04 | 0.4688 | 0.4270 | 0.3998 |
| NCS02 | 0.4852 | 0.4314 | 0.4037 |

Finally, the runs GATE01 and GATE02 using the H-SVM obtained much worse results than the other two runs GATE03 and GATE04 which used the SVM for flat classification. That may be due to the specific way we used for computing the confidence score for every F-term. We will come back to this problem later. On the other hand, the evaluation measure used in the NTCIR-6 evaluation does not count the partial matches between two closely related classes, which may occur more frequently in the results of the H-SVM than for the flat SVM classification, due to their different mechanisms.

Table 2 presents the results of the five runs in the Table 1, using a new evaluation measure which counted the exact matches as well as the partial matches. According to the evaluation measure, the system obtained a score $1$ for one exact match and a score between $0$ and $1$ for one partial match. The exact score for one partial match was dependent upon the cost between the true class and the predicted class in the partial match – the higher the cost is, the lower score the partial match obtains. For the detailed description of the new measure see our another paper [4]. We can see that, when using some new evaluation measure which takes into account the relations between the class labels, the gap of the results between the H-SVM and the flat SVM become significantly narrower. For example, if we only consider the exact matches, the A-Precision of the H-SVM was less than half of that of the flat SVM. But if we consider the both exact and partial matches, the A-Precision of the H-SVM was about 80% of that of the flat SVM. Hence, comparing the results counting the exact matches only with those counting both the exact matches and the partial matches, we can see that, if an instance cannot be classified correctly, the H-SVM has more tendency than the flat SVM to classify the instance into a class which is close to the true class. However, unfortunately, even using the new evaluation measure, the performance of the H-SVM was still worse than the flat SVM.

**Table 2. Results by using a new evaluation measure which took into account the exact matches as well as the partial matches.**

| Run-ID | A-Precision | R-Precision | $F_1$ |
|---|---|---|---|
| GATE01 | 0.5193 | 0.5414 | 0.3605 |
| GATE02 | 0.4864 | 0.4982 | 0.3479 |
| GATE03 | 0.6269 | 0.6194 | 0.4429 |
| GATE04 | 0.6221 | 0.6138 | 0.4363 |
| NCS02 | 0.6463 | 0.6241 | 0.4363 |

## 4.2 Results for different settings

Our system is based on the SVM learning algorithm. However, it was not a straightforward application of SVM to F-term classification. Instead, as discussed above, we have employed several techniques to adapt the SVM to the task. After submitting our results for official evaluation, we carried out experiments to evaluate the techniques and the features used in our system. In those experiments we used the same training and testing data as in the official run of the task but different experimental settings. In the follows we presents the experimental results.

First we would like to compare the results of using different text of the patent document. Our four runs have showed clearly that using the short F-term description can boost the performance. Table 3 presents the results of using only the abstract section of patent document. The other experimental settings were the same as the GATE03 runs. Hence we can compare the figures with those of the GATE03 in Table 1. We can see that using the abstract of the patent obtained much worse results than using the full content of patent.

**Table 3. Results using the abstract section of patent document only. The other settings were the same as those of the GATE03 run.**

| A-Precision | R-Precision | $F_1$ |
|---|---|---|
| 0.4279 | 0.3908 | 0.3647 |

Secondly, our system did not use the standard SVM algorithm. Instead it used the uneven margins SVM, which often achieved much better F-measure score than the standard SVM on the imbalanced data where the negative example outnumbered the positive examples. In comparison with the standard SVM model which treats the positive examples and negative example equally, the uneven margins SVM used an uneven margin parameter $\tau$ to adjust the ratio of positive margin to negative margin of the learned classification hyper-plane in the feature space. See [5] for detailed description of the uneven margins SVM. In our submitted runs we set the uneven margins parameter $\tau$ as $0.5$. Note that $\tau = 1.0$ leads to the standard SVM model. From Table 4 we can see that the uneven margins SVM obtained clearly higher $F_1$ value than the standard SVM model.

Note that we used the same value of the uneven margins parameter for all the SVM models, which was equivalent to the same shift of the confidence scores for all the SVM models, and the same shift of the scores for all the SVM model did not change the rank order of those scores. Therefore the A-Precision and R-precision of the uneven margins SVM model were the same as those of the standard SVM.

**Table 4. Comparison between the standard SVM ($\tau = 1.0$) and the uneven margins SVM ($\tau = 0.5$).**

| $\tau$ | Precision | Recall | $F_1$ |
|---|---|---|---|
| 1.0 | 0.5479 | 0.3107 | 0.3643 |
| 0.5 | 0.4075 | 0.4904 | 0.4125 |

Thirdly, we normalised the weight vector of the SVM model to facilitate the comparisons of the scores from different SVM models for one test document. Table 5 presents the results without the weight vector normalisation of the SVM model. In comparison with the results of the GATE03 with the normalisation presented in Table 1, the results without normalisation became worse. But the difference was not as big as we expected, in particular for the R-Precision.

**Table 5. Results for the SVM model without the normalisation of the SVM weight vector. The other settings were the same as those of the GATE03 run.**

| A-Precision | R-Precision | $F_1$ |
|---|---|---|
| 0.4643 | 0.4330 | 0.3677 |

Finally, we discuss some experimental results of the H-SVM. As said in Sub-section 3.2 about the run GATE02, we could use two different methods to obtain the score of one particular F-term. One method was to use the score of the F-term classifier itself. Another was the average of the scores of the SVM classifiers of the F-term itself and all the ancestor F-terms. In our submitted runs GATE01 and GATE02 we used the second method. Table 6 presents the results using the first method, which were much lower than the corresponding results of the GATE01 run (listed in Table 1) using the second method.

**Table 6. Results of the H-SVM using the score of the F-term classifier itself. The other settings were the same as those of the GATE01 run.**

| A-Precision | R-Precision | $F_1$ |
|---|---|---|
| 0.1621 | 0.1188 | 0.1248 |

Recall that in our submitted runs GATE01 and GATE02 for the H-SVM, in order to decide if a patent has one particular F-term, we used the averaged score of the SVM classifiers of the F-terms along the path from the top F-term to the F-term considered. Actually we could used a different method to make such decision, by which we assign one F-term to one patent if and only if all the SVM classifiers from the top F-term down to the F-term itself classify the patent as positive example, and the classifier of any child F-term of the current F-term, if there is any, classifies the patent as negative example. We call this method as H-score method. Table 7 shows that H-score method had much higher Precision but somehow lower Recall and a higher $F_1$ score.

**Table 7. Comparisons of the F-measure results for the H-SVM between the averaged score and H-score.**

| | Precision | Recall | $F_1$ |
|---|---|---|---|
| Averaged score | 0.1488 | 0.6374 | 0.2257 |
| H-score | 0.3484 | 0.4956 | 0.3840 |

## 5 Conclusions and discussions

Our SVM-based learning system has obtained very good results on the F-term classification subtask at NTCIR-6 Patent Retrieval Task. It achieved the best results according to two of the three measures used in the task evaluation, namely the R-Precision and F-measure. We adopted several techniques to adapt the SVM algorithm to the F-term classification problem. The additional experimental results showed that our system does indeed benefit from these adaptations. Our system also benefited from the full patent text in addition to using the F-term descriptions as extra training material.

However, we were somewhat surprised that H-SVM, which takes into account the hierarchical relations among F-terms under each patent theme, obtained much worse results than the flat SVM classification. Using new evaluation measure, which counted both exact matches and partial matches, showed that H-SVM indeed tended to minimuse errors by classifying the patents into the F-term which is closer to the true F-term, in cases when it could not classify the patent with the correct F-term. However, H-SVM's capability for correct classification seems much worse than that of the flat SVM, which led to poor overall performance of the H-SVM system. One possible reason for the low results is due to the fact that H-SVM learning is dependent upon the hierarchical relations among the classes, but these were much too complicated for the H-SVM to get an appropriate classification score for the test instances, as we demonstrate in our additional experimental results discussed in the previous sections.

On the other hand, it is worth noting that the F-term classification problem has some unique characteristics in comparison to the standard hierarchical classification task, which might also contribute to H-SVM's low performance.

First, the F-terms under a given theme are not hierarchically related with each other in the strict sense, because, as pointed out in [2], some middle F-term (namely not the leafy F-term in the F-term hierarchy) represents two different things — one is all sub-elements not considered by its child elements, and another one is the parent element of its all child elements. Hence, it would be helpful to the hierarchical learning algorithm if the middle F-term can be split as two nodes representing the two different meaning respectively. By doing so the F-terms will have a more proper hierarchical relations.

Secondly, the F-term classification is a multi-class problem in the sense that each instance often has more than one true classes. On the other hand, H-SVM was designed for the problem in which each instance has only one true class so that one test instance can be classified from the top class down to the bottom class. If one instance has more than one true classes, it is impossible for the binary SVM classifier, corresponding to one of those common ancestor classes of the two true classes, to classify the instance correctly into the two paths that contain the two true classes respectively.

The above discussions may give some insights into the reasons for H-SVM's poorer results on the F-term classification when compared to the flat SVM, despite the fact that H-SVM has obtained previously better results on other hierarchical classification tasks (see [1] and [7]). On the other hand, it is worth investigating further the application of hierarchical classification algorithms to patent F-term classification, since the state of the art results is not good enough and the specific/general relations among the F-terms should be useful for the hierarchical classification algorithms.

## 6 Acknowledgements

This work is supported by the EU-funded SEKT (IST-2004-506826) and KnowledgeWeb (IST-2004-507482) projects.

## References

[1] N. Cesa-Bianchi, C. Gentile, A. Tironi, and L. Zaniboni. Incremental Algorithms for Hierarchical Classification. In *Neural Information Processing Systems*, 2004.

[2] M. Iwayama, A. Fujii, and N. Kando. Overview of Classification Subtask at NTCIR-5 Patent Retrieval Task. In *Proceedings of NTCIR-5 Workshop Meeting*, 2005.

[3] T. Joachims. Text categorization with support vector machines: learning with many relevant features. In C. Nédellec and C. Rouveirol, editors, *Proceedings of ECML-98, 10th European Conference on Machine Learning*, number 1398 in Lecture Notes in Computer Science, pages 137–142, Chemnitz, DE, 1998. Springer Verlag, Heidelberg, DE.

[4] Y. Li, K. Bontcheva, and H. Cunningham. New Evaluation Measures for F-term Patent Classification. In *The First International Workshop on Evaluating Information Access (EVIA 2007)*, 2007.

[5] Y. Li and J. Shawe-Taylor. The SVM with Uneven Margins and Chinese Document Categorization. In *Proceedings of The 17th Pacific Asia Conference on Language, Information and Computation (PACLIC17)*, Singapore, Oct. 2003.

[6] M. Makita, S. Higuchi, A. Fujii, and T. Ishikawa. A system for Japanese/English/Korean multilingual patent retrieval. In *Proceedings of Machine Translation Summit IX (online at http://www.amtaweb.org/summit/MTSummit/papers.html)*, Sept. 2003.

[7] J. Rousu, C. Saunders, S. Szedmak, and J. Shawe-Taylor. Learning Hierarchical Multi-Category Text Classification Models. *Journal of Machine Learning Research*, 7:1601—1626, 2006.