CLEF-IP 2011 Working Notes: Utilizing Prior Art Candidate Search Results for Refined IPC Classification

Hyung-Kook Seo, Kyouyeol Han, Jaean Lee

2F Ottogi Center, 1009-1 Daechi-Dong,
Kangnam-Gu, Seoul, Korea
{hkseo, sept102, jalee}@wisenut.co.kr

Abstract. For the refined IPC classification in the CLEF-IP 2011 task, we constructed classification system with KNN classification which uses PAC (Prior Art Candidate) search results as neighbors. We also slightly modified the neighborhood evaluation. We also furnished a simple PAC search system. We produced some running results both in PAC search and classification, and evaluated our system. Our test showed an improved result in the refined IPC classification.

Keywords: Prior Art Candidate Search, IPC Classification, Document Categorization, KNN Classification

1 Introduction

Our lab performed prior art candidate (PAC) search and IPC classification for the Korean Intellectual Property Office. We dealt with domestic patents in Korea, so linguistic analysis was limited to Korean. However PACs are not limited to domestic patents, and we also have interests in the PAC search of other languages, including English and Japanese.

We decided to participate in CLEF-IP to share our experience in patent domain and extend our technical coverage and experience to other patents domain like European patent corpus.

However, we have limited knowledge about European patent structure (except commonly shared fields like claims, IPCs, etc.) and lack experience in linguistic analysis of other languages (except Korean). We also were on a limited time schedules, so we focused our interests on refined IPC classification. But our approach needs PAC results, so we also implemented our PAC system.

1.1 Test Collection and Topics

Like other participants in CLEF-IP 2011, we used only test the data collection which comprises extracts of the MAREC dataset by IRF.

We only used this collection in the entire process for producing running result of the tasks we participated.

1.2 CLEF-IP Tasks Participated

We participated in following three tasks in CLEF-IP 2011 [1]:

1. Prior Art Candidate (PAC) Search
2. IPC Classification: up to subclass level
3. Refined IPC Classification: up to subgroup level, with given subclass value

As we stated before, our ultimate interests lie in *refined IPC classification*, and there were small efforts to improve PAC search results. This is described in the next chapter.

2 Approaches to Refined Classification

Classification of a patent up to sub-class degree is quite difficult task for model-based classification, because of sparseness of training samples in that level. So, we implemented indirect (and simple) method, that implements KNN-like classification using PAC search results.

2.1 Existing Classification Approach

In patent classification, Kostar et al. [2] proposed a method using the winnow algorithm. Winnow is a mistake driven learning algorithm that computes for each category a vector of weights for separating between relevant and irrelevant patent [3]. In this study, they obtained F-measure of around 68% (multi-categorization). This result was measured with a customized success criterion and relatively few documents.

Fall et al. [4] applied various machine learning algorithms to patent classification. The machine learning algorithms were Naive Bayes(NB), SNoW, support vector machines(SVM) and k-nearest neighbor(KNN) algorithms. Here SNoW is a variation of the winnow algorithm. They investigated useful patent document fields to index, and defined three measures of categorization success. As a result, they presented the best precision of 41%(SVM), 39%(NB), 33%(NB) and 36%(SNoW) when the first 300 words are indexed at subclass level. In first three guesses KNN achieved the best precision of 62% and all categories SVM achieved the best precision of 48%. They [5] also presented a customized language-independent text classification system for categorization.

When the amount of training data increses, a model-based system has increased feature scale and time complexity. In order to reduce the feature scale, some of researchers limited the number of documents, term selection, and length of the documents. To reduce time complexity, other have attempted instance-based learning such as KNN. It first selects K samples when the similarity values are sorted in descending order, and then determines the categories of test sample with class mapping method. It makes a trade-off between effectiveness and time complexity.

W. Wang et al.[6] reported their experience in the NTCIR-7 Patent Mining Task(MT) to classify patent documents according to the IPC taxonomy. Their

approach is based on the KNN algorithm using cosine and Euclid distance similarity. And T. Xiao et al.[7] described their methodology that are used KNN and re-ranking models. They achieved a mean average precision (MAP) of 48.86% when classifying according to the subgroup level. Also [8] reported result of their experiments on the automatic assignment of patent classification to research paper abstracts. The results showed the best precision of 50.62% (MAP) when using formal run data and particular query group.

Lately, Y. Cai et al. [9] presented a KNN text categorization method based on shared nearest neighbor, effectively combining the BM25 similarity calculation method and the Neighborhood Information of samples in the NTCIR-8 workshop. BM25 is a bag-of-words retrieval function, combines the word frequency and document frequency, balances the length of the document, and is a highly efficient similarity calculation method. They conducted a comparison experiment on Japanese corpus and English corpus provided by the National Institute of Informatics from 1993 to 2002, using the basic KNN and KNN based on shared nearest neighbors. Compared to KNN method, KNN+SNN method showed 72.12% precision (about 0.03) higher at subclass levels and 36.93% precision at subgroup levels on English corpus.

2.2 Our IPC Classification Approach

As we know about KNN Classification, K nearest neighbors are K documents most similar with the given query document to be classified [10].

So it can be easily connected with search results. That is, top K search results with query document can be directly adopted in KNN classification, and one system used this simple method, though it was not so competitive with model-based classification algorithms [11].

In fact, at subgroup level we need about 70,000 categories to be trained, and most classification models suffer from sparseness of training documents and problems in system memory (for loading models) and processing time (training or classification itself). But according to our experience in Korean patent domain, KNN classification with PAC search showed quite good quality in classification of subgroup level.

So we tried to construct refined IPC classification system utilizing PAC results. (And IPC classification up to subclass level as well. In fact, we paid not so much attention in the optimization or improvement in subclass level, because of limited time)

2.3 PAC Search Approach

We implemented a PAC search system using only selected weighted keywords which are extracted from major content fields (title of invention, abstract, description, claims).

We added two additional efforts in PAC search to improve our results. They are the following:

Removing Non-Content Words

A Document will be represented by a set of words which consist of content-word and functional word. After POS tagging, we try to remove functional words and stop word. While stop words are controlled by human input and cannot be automated, we can algorithmically find words which don't describe a particular document. (non-content words).

The standard probabilistic model for the distribution of a certain type of event over units of a fixed size is the Poisson distribution.

$$\text{Poisson Distribution: } p(k;\ \lambda_i) = e^{-\lambda_i}\frac{\lambda_i^k}{k!} \text{ for some } \lambda_i > 0 \qquad (1)$$

The most common model of the Poisson distribution in IR, the parameter $\lambda_i > 0$ is the average number of occurrences of w_i per document: that is, $\lambda_i = \frac{cf_i}{N}$ where cf_i is the collection frequency and N is the total number of documents in the collection. And we can get an approximation of DF by $N\left(1 - p(0;\lambda_i)\right)$. As this model assumes independence between term occurrences, its estimations are good for non-content words. [12]

Algorithm:

1. Calculate λ_i: collection frequency of i / N (N is total number of document in the collection)
2. Calculate expected document frequency by Poisson distribution: $N(1 - p(0;\lambda_i))$
3. Get the overestimation value: expected document frequency(i)/df(i)

Parameter:

1. Document Frequency Rate: Percentage of the number of document in the collection
2. Lower overestimation criteria and upper overestimation criteria

Result:

According to parameter tests, our system (in Korean) showed best result under observed document frequency of 0.05%, lower overestimation criteria of 0.9 and upper overestimation criteria of 1.0.

In the CLEF-IP 2011, we'll fix lower overestimation criteria as 0.9, and change upper overestimation criteria from 1.0 to 1.3.

Extracting Co-Occurrence Terms

A null hypothesis is often stated by saying the parameter Θ is in a specified subset Θ_0 of the parameter space Θ.

$$H_0: \theta \in \Theta_0$$
$$H_1: \theta \in \Theta_0^C \qquad (2)$$

Likelihood ratios are an approach to hypothesis testing. The likelihood function is $L(\theta|x) = f(x|\theta)$ is a function of the parameter θ with x held fixed at the value that was actually observed, i.e., the data. The likelihood ratio test statistic is [14]

$$\text{Likelihood Ratio: } \Lambda(x) = \frac{\sup\{L(\theta|x): \theta \in \Theta_0\}}{\sup\{L(\theta|x): \theta \in \Theta\}} \quad (3)$$

In applying the likelihood ratio test to collocation discovery, we examine the following two alternative explanations for the occurrence frequency of a bigram w_1w_2 [13]

$$H_0 = P(w^2|w^1) = p = P(w^2|\text{not } w^1)$$
$$H_1 = P(w^2|w^1) = p_1 \neq p_2 = P(w^2|\text{not } w^1) \quad (4)$$

We used the usual maximum likelihood estimates for p, p_1 and p_2 and write c_1, c_2, and c_{12} for the number of occurrences of w_1, w_2 and w_1w_2 in the corpus:

$$p = \frac{c_2}{N}, \quad p_1 = \frac{c_{12}}{c_1}, p_2 = \frac{c_2 - c_{12}}{N - c_1} \quad (5)$$

Now we can get likelihood ratio by assuming a binomial distribution and then following is asymptotically X^2 distributed

$$-2 \times \log\frac{L(H_0)}{L(H_1)} \quad (6)$$

In the CLEF-IP 2011, we used confidence level of α=99.9. We tried both with co-occurrence terms and without them in the runs.

3 System Setup

We implemented our system according to procedures to be explained in this chapter.

We first extracted weighted keywords from each patent xml file provided in CLEF-IP 2011 corpora, combined them in several bulk files, and indexed them. For indexer and searcher, we used Lucene. We implemented a simple searcher program implemented in Java, and a final classifier program applying search results.

3.1 Overall Architecture

Our system is illustrated in the following architecture diagram. It's very similar with traditional document search system.

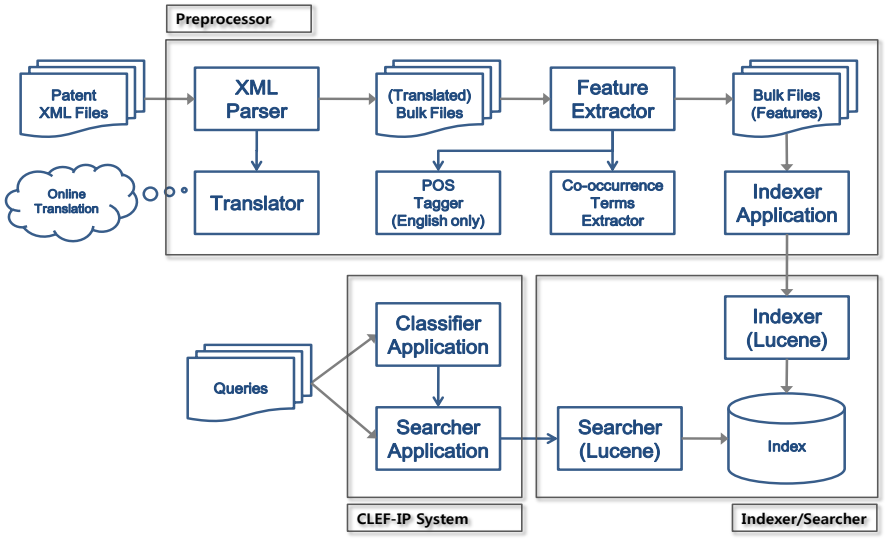

Fig. 1. Entire system architecture.

We used our in-house English POS tagger for base English analysis.

For translating other languages like French or German into English, we used open online translation service, *MyMemory* [15].

We used *Lucene 3.1.0* for the base search engine, and for accessing this engine, we wrote simple java applications for indexing and searching. We also wrote an application for classification which calls the searcher application.

3.2 Preprocessing and Indexing

Basically, we used a nearly identical preprocessing system except this time we used English POS tagger instead of Korean one.

We selected only one XML document among various versions of a same patent to guarantee uniqueness of the patent, so that there's no patent document with same application number in the entire index system. After this process, we got 1,331,182 unique patents in the EP(European Patents), and 437,987 in the WO(WIPO Patents)

During this procedure we also translated content fields with the online translation service. After some sample runs, we discovered that translating full text would consume a lot of time (and may lead to missing the CLEF-IP 2011 deadline), so we only translated the abstract and select sentences(about 2048 characters) in other content fields.

For feature extraction, we used these content fields: Title of Invention, Abstract, Description, and Claims. We also extracted some co-occurrence terms and select up to 5 terms with extracted features.

We finally produced bulk files with features with and without co-occurrence terms (we call them *co-terms* for simplification) for indexing. And we produced two separated indices, to analyze the effect of co-terms used in the search.

We also preprocessed patents used as topics (queries). In this case, translation with full content was conducted.

3.3 Prior Art Candidate Search

We produced a total of 8 runs for search results. First 4 runs target index without co-terms, and other 4 runs target index with co-terms. In each group, we changed upper overestimation threshold for non-content words removal from 1.0 to 1.3 (1.0, 1.1, 1.2, 1.3) resulting in 4 runs for each group.

We produced 1,000 results for every patent query. The results are produced without lower threshold in the weight of search results, so in most cases, our search results per one document were almost 1,000 documents.

3.4 Classification

Because we used search results of PAC search, we also produced 8 runs for each classification task.

For KNN classification, we set K as 1,000, because we produced 1,000 search results per a query.

In fact, we have observed that combining reciprocal of search results than just counting the number of patents per category shows much better results. It's similar to adoption of weighting in the average precision [16]. We basically adopted this improved weighting scheme in the KNN classification results we got.

To verify this intuition, we also ran the base condition and compared this result with improved weighting scheme applied in the CLEF-IP 2011 runs.

4 Results and Analysis

Following is a simple report on our results.

4.1 PAC Search Results

We simply show the result of our runs along with best runs of CLEF-IP 2011.

Table 1. PAC search results compared with best runs of other CLEF-IP 2011 participants.

RUN_NAME	Entire Language						English Only					
	MAP	SET_P	SET_recall	recall_5	recall_10	recall_20	MAP	SET_P	SET_recall	recall_5	recall_10	recall_20
CHEMNITZ.CUT_UHI_CLEFIP_BOW	0.0914	0.0037	0.4318	0.0896	0.1251	0.1635	0.1009	0.0049	0.5233	0.0956	0.1401	0.1921
HYDERABAD. TEXTRANK_IDFCITATIONALL	0.097	0.0036	0.3993	0.0932	0.118	0.1489	0.0943	0.0046	0.482	0.0897	0.1237	0.1671
WISENUT_R1_BASE_PAC	0.0565	0.0028	0.3948	0.0562	0.081	0.1125	0.0836	0.0036	0.4677	0.0812	0.1137	0.1573
WISENUT_R2_BASE_10_PAC	0.0566	0.0028	0.3949	0.0563	0.081	0.1125	0.0836	0.0036	0.468	0.0812	0.1137	0.1573
WISENUT_R3_BASE_30_PAC	0.0566	0.0028	0.3949	0.0563	0.0811	0.1126	0.0837	0.0036	0.468	0.0813	0.1137	0.1574
WISENUT_R4_BASE_30_PAC	0.0567	0.0028	0.3949	0.0565	0.0811	0.1125	0.0837	0.0036	0.468	0.0817	0.1138	0.1574
WISENUT_R5_CO_PAC	0.0573	0.0028	0.3966	0.0564	0.0809	0.1112	0.0841	0.0036	0.4656	0.0803	0.1126	0.1535
WISENUT_R6_CO_10_PAC	0.0573	0.0028	0.3966	0.0564	0.081	0.1112	0.0841	0.0036	0.4656	0.0803	0.1126	0.1535
WISENUT_R7_CO_20_PAC	0.0573	0.0028	0.3966	0.0564	0.081	0.1114	0.0841	0.0036	0.4658	0.0802	0.1127	0.1538
WISENUT_R8_CO_30_PAC	0.0573	0.0028	0.3966	0.0564	0.0808	0.1115	0.0841	0.0036	0.466	0.0803	0.1123	0.1538

We got slightly improved result when co-terms are applied. And the differences in upper threshold in the non-content words extraction made no special differences.

And due to multilingual issues, our result showed quite low quality. (It's partially displayed in English results that show quite narrower gaps with the top runners) We'll try to find alternatives to overcome this issue.

4.2 IPC Classification Results

We also got the results of our classification runs and compared them with ones from the other participant.

Table 2. IPC classification results compared with other CLEF-IP 2011 participant.

RUN_NAME	set_P	set_recall	set_F_1.0
NIJMEGEN.RUN_ADMWCIT_CLS1	0.5379	0.8563	0.6168
NIJMEGEN.RUN_ADMW_CLS1	0.5436	0.8506	0.6186
WISENUT.WISENUT_R1_BASE_CLS1	0.2867	0.838	0.4021
WISENUT.WISENUT_R2_BASE_10_CLS1	0.2871	0.8389	0.4027
WISENUT.WISENUT_R3_BASE_20_CLS1	0.2869	0.8384	0.4024
WISENUT.WISENUT_R4_BASE_30_CLS1	0.2871	0.8387	0.4027
WISENUT.WISENUT_R5_CO_CLS1	0.2882	0.8366	0.4027
WISENUT.WISENUT_R6_CO_10_CLS1	0.2883	0.8371	0.4029
WISENUT.WISENUT_R7_CO_20_CLS1	0.2885	0.8376	0.4032
WISENUT.WISENUT_R8_CO_30_CLS1	0.2884	0.8376	0.4031

Because we do not use model-based methods, our result showed lower result in the precision. We also didn't limit the score of classification result; if we tune the score thresholds, it's expected that we may produce much better results.

4.3 Refined IPC Classification Results

Finally, our refined IPC classification results are displayed in the table below. If more labs participated in this track, we may get a better perspective on our quality.

Table 3. Refined IPC classification results compared with other CLEF-IP 2011 participant.

RUN_NAME	set_P	set_recall	set_F_1.0
NIJMEGEN.RUN_WINNOW_WORDS_CLS2	0.0731	0.0622	0.0609
WISENUT.WISENUT_R1_BASE_CLS2	0.2928	0.495	0.3326
WISENUT.WISENUT_R2_BASE_10_CLS2	0.293	0.4951	0.3327
WISENUT.WISENUT_R3_BASE_20_CLS2	0.293	0.4952	0.3328
WISENUT.WISENUT_R4_BASE_30_CLS2	0.2928	0.4954	0.3328
WISENUT.WISENUT_R5_CO_CLS2	0.2925	0.494	0.332
WISENUT.WISENUT_R6_CO_10_CLS2	0.2926	0.4938	0.3319
WISENUT.WISENUT_R7_CO_20_CLS2	0.2925	0.4938	0.3319
WISENUT.WISENUT_R8_CO_30_CLS2	0.2926	0.4943	0.3321

Whie simple comparison is quite dangerous, our system showed quite improved results in this track.

And as we stated before, we compared the new, refined weighting scheme (which is applied in CLEF-IP 2011) with the base one. Following table shows that result.

Table 4. Refined IPC classification results comparison between weighting schemes.

Classification Results	Scheme with Rank				Base Scheme			
	P	Recall	F1	MAP	P	Recall	F1	MAP
up to 1 result	0.4453	0.1893	0.2657	0.2476	0.2397	0.1019	0.1430	0.1241
up to 5 results	0.2251	0.4606	0.3024	0.4028	0.1470	0.3018	0.1977	0.2202
up to 10 results	0.1473	0.5791	0.2348	0.4297	0.1064	0.4204	0.1698	0.2450
up to 20 results	0.0934	0.6869	0.1645	0.4433	0.0747	0.5480	0.1314	0.2598

Refined scheme showed far better results than the base scheme, especially in precision. MAPs were also dramatically improved. (Note that precision and recall were micro-averaged, so they're quite different from our reported values)

Considering the result of [9], our result is very promising, because precision in suggesting one IPC classification result showed almost the same or improved quality.

5 Conclusion and Future Work

We implemented a simple refined IPC classification system utilizing search results provided from PACS system. Though our PACS system showed rather lower performance than those of other labs, our refined classification results based on the search results of our system showed quite good performance, especially when the criteria for category selection is changed.

We left some challenges as future work.

First, we can improve PACS search results. For example, we didn't set threshold in the score, only the maximum number of results. And we had a major problem in search results due to multilingual defects of our system. We may improve these problems at the next workshop.

Second, we can adapt model-based classification up to subclass level. In fact, it's true that model-based classification method works well up to subclass level, so our IPC classification system should use classification model like SVM does.

Finally, we may optimize weighting factor of ranked documents in the refined IPC classification. As just using reciprocal of ranks in the search results improved the refined classification, it's expected that adopting more sophisticated weighting factor in the KNN can produce improved classification results.

References

1. Piroi, F.: CLEF- IP 2011: Track Guidelines. IRF, Vienna (2011)
2. Koster, C.H.A., Seutter, M., Beney, J.: Classifying Patent Applications with Winnow. In: Proceedings Benelearn Conference, Antwerpen (2001)
3. Littlestone, N.: Learning Quickly when Irrelevant Attributes Abound: A new Linear-Threshold Algorithm. In: Machine Learning, Vol. 2, pp. 285--318. Springer, Nethelands (1988)
4. Fall, C. J., Torcsvari, A., Benzineb, K., & Karetka, G.: Automated Categorization in the International Patent Classification. In: ACM SIGIR Forum, Vol. 37, Issue 1. ACM, New York (2003)
5. Fall C. J., Benzineb K., Guyot J., Torcsvari A., Fievet P.: Computer-Assisted Categorization of Patent Documents in the International Patent Classification. In: Proceedings of the International Chemical Information Conference (ICIC'03), Nimes, France (2003)
6. Wang, W., Li, S., Wang. C.: ICL at NTCIR-7: A Improved KNN Algorithm for Text Categorization. In: Proceedings of NTCIR-7 Workshop Meeting (2008)
7. Xiao, T., Cao, F., Li, T., Song, G., Zhou, K., Zhu, J., Wang, H.: KNN and Re-ranking Models for English Patent Mining at NTCIR-7. In: Proceedings of NTCIR-7 Workshop Meeting (2008)
8. Mase, H., Iwayama, M.: Hitachi Ltd.: NTCIR-7 Patent Mining Experiments at Hitachi. In: Proceedings of NTCIR-7 Workshop Meeting (2008)
9. Cai, Y., Ji, D., Cai, D.: A KNN Research Paper Classification Method Based on Shared Nearest Neighbor. In: Proceedings of the 8th NTCIR Workshop Meeting. pp. 336--340. (2008)
10. k-nearest neighbor algorithm, http://en.wikipedia.org/wiki/K-nearest_neighbor_algorithm
11. Lopez, P., Romary, L.: Experiments with Citation Mining and Key-Term Extraction for Prior Art Search. In: CLEF-IP 2010, Padua (2010)

12. Manning, C.D., Schütze, H.: Foundations of Statistical Natural Language Processing, MIT Press, Cambridge (1999)
13. Dunning, T.: Accurate Methods for the Statistics of Surprise and Coincidence. In: Computational Linguistics 19. pp. 61--74. (1993)
14. Casella, G., Berger, R.L.: Statistical Inference, 2nd edition, p. 375. Duxbury Press (2001)
15. MyMemory, http://mymemory.translated.net
16. Average Precision, http://en.wikipedia.org/wiki/Information_retrieval#Average_precision