IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 16, Issue 6, Ver. IV (Nov – Dec. 2014), PP 07-11
www.iosrjournals.org

An Efficient Model for Patent Search Pattern

[1]B. Harika, [2]Mr. M.V.Jagannathareddy
[1]M.Tech(CSE),MITS,MADANAPALLE [2]Associate Professor in CSE Dept,MITS,MADANAPALLE

Abstract:*Patents are very important for protecting the sophisticated property of individuals.Patent Search helps us to find the previously developed similar patents and used to certify the newly developing patent applications. As such it has gained both economical and technical attention. In the Existing system there is no proper attention to user's search purpose and searching time responsiveness. It returns answers only if the given query has matches and if not we should repeatedly submit the questions and wait for answers. But, our Proposed System overcomes all this disadvantages. In the proposed system we use error-correction, topic-based query submission and query extension. We use to rank the patents according to the best use of the patents. We use a new method to improve the search efficiency by classifying the patents according to the type of topics of interest. So that users can interactively submit the questions and get the relative answers. We also provide the users gratifications of the patents searched by the users so that they can know the highly searched patent. In future our proposed system can be implemented by connecting large number of database.*

I. Introduction

PATENTS are very important because they act as the security to the property of an individual or a company. Patent search is also very important as to know the recently published patents and also to know the valid and invalid patent applications. There are many online systems that support the patent search like Google Patent Search, Derwent Innovations Index, USPTO but these doesn't have much knowledge about the new patent applications.

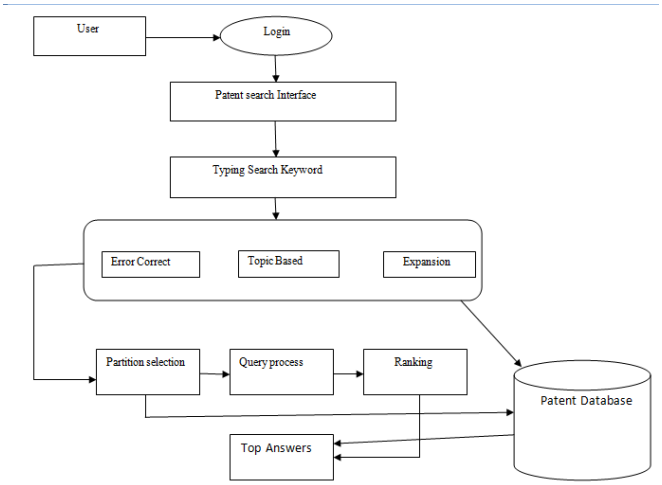
In the Existing system the user has to repeatedly issue the queries until he has given the correct query with spelling. The user has to search for all the topics until he gets the desired topic as they all are implemented in single database. No Ranking is used to display the highest ranked patents.

To make the users search effective the first step is to put the patents into topic-wise and also to suggest the correct spelling to the user even if he enters the mispelt query then the users intention will be gained. For Example, if the user types Hundai instead of Hyundai the existing system will not display any result as it doesn't match with the keyword they have. The existing systems do not gain any user attention due to the lack of these services. So, We Proposed this efficient model for patent search.

II. Architecture Of Our Efficient Model For Patent Search

Our Proposed system An Efficient model for patent search pattern helps the users to easily search their desired patents.

The below figure depicts the Architecture of the Patent Search Model. This architecture consists of three components Error-Correction, Topic-based query suggestion and Topic expansion. These techniques helps the user to easily search for the patents and get their desired patents quickly. To improve the efficiency we divide the patents as domain and sub-domain where the domain suggest the name of the type of the patent and sub-domain suggests the name of the product.



III. Efficient Approach for Patent Search

To understand the user's query intention and to match the query entered by the user and the keywords of the patents is a very difficult task. So we present the following techniques to help the users easily find their related patents,

3.1 Dividing the Patents

We divide the patents due to the following reasons:- First the patents are referred as classes and there are around 500 classes and 136000 subclasses. The patents are also large in number. In USPTO the number of patents is 6 million patents and 4 million patent applications for approval. The number of patents are increasing at the rate of 30percent every year in japan also only some classes / subclasses are relevant to the query given by the user. For these reasons we divide the patents into Domain and Sub-domain where the Domain consists of the type of the Patent and Sub-domain consists of the name of the particular product and also the sub-domains are further divided into large numbers based on the products.

3.2 Error-Correction

When the user enters the query sometimes it may contain errors like spelling mistakes and mismatch occurs between the query entered by the user and the patent name. The existing systems do not answer for the query in such situations as they could not find the relevant patents. But, the best solution for this is to automatically correct the mispelt keyword entered by the user and return the answers to the user. The edit distance is the best method for this purpose. The edit distance is nothing but the minimum number of the edit operations like insertion, deletion and substitution of one or more characters to correct the user entered query. For Example:- the edit distance between Hunday and Hyundai is 2. Two keywords are said to be similar if their edit distance is 0. There are other methods for effective error-correction like filter and refine method. This method first use filtering techniques to see whether the keywords entered by the user and the patent name are matching if not then it will go the verification stage where the wrong spelt words are corrected and return the similar keyword to the user. In our proposed system the query is checked letter by letter and it is not matched in a particular domain then it will search in another domain until it is found then it checks in the respective sub-domains and if there are any mistakes in the query entered by the user then the similar keywords or topics are displayed to the user so that this will be very helpful for the user even if he enters the wrong or misplet keyword or query.

3.3 Topic-based Query

An LDA model is used to suggest the topics to the users based on their search. An LDA model is a soft-clustering technique where the keyword is present in multiple topics and the degree of each keyword is taken into account. The keyword is distributed along several topics by using the language model. This model predicts the likelihood of sampling a particular keyword. Thus, by combining these two probabilities we can suggest the related topics to the users based on their search.

Let $s=q_1,q_2,\ldots\ldots,q_{|s|}$ be the set ofsequence of previous queries. Let p be the prefix of the term entered by the user. $C=\{c_1,c_2,\ldots..,c|C|\}$ be the set of complete terms with prefix p. $D=\{d_1,d_2,\ldots.,d_{|D|}\}$ represent the number of partitions that the patents are dividedand $T=\{t_1,t_2,\ldots..,t_{|T|}\}$ represent the set of topics. Then the probability of P(c/s) which represents the top terms with highest probabilities is defined as,

$P(c/s)= \sum_{ti\epsilon T} \sum_{dj\epsilon D} P(c/s; t_i, d_j) \,.\, P(t_i\,, d_j/s)$

$$P(c|s) = \sum_{t_i \in T} \sum_{d_j \in D} P(c|s; t_i, d_j) \cdot P(t_i, d_j|s) \text{ [4].}$$

Here, we estimate the likelihood that a user issues a keyword c if t_i is the topic of c and $d_j$returns the partition number where the keyword presents.

$$P(c|s) = \sum_{t_i \in T} \sum_{d_j \in D} \left(\lambda P(c|t_i) + (1-\lambda) P(c|d_j)\right) P(t_i|s) P(d_j|s)$$
$$= \lambda \sum_{t_i \in T} P(c|t_i) P(t_i|s) + (1-\lambda) \sum_{d_j \in D} P(c|d_j) P(d_j|s) \text{ [4]}$$

$$P(t_i|s) = \frac{P(s|t_i) \cdot P(t_i)}{P(s)} \text{ [4].}$$

$$P(c|d_j) = (1-\eta) \frac{\text{count}(c, d_j)}{|d_j|} + \eta \frac{\text{count}(c, D)}{|D|} \text{[4],}$$

3.4 Query Expansion

When the user enters the query in the existing system there is no query expansion that is when the user searches for a particular product patent then all the other products related to that patent are not displayed. For Example if the user wants to know the patent right of a car it displays only the details regarding the company that releases that car but it will not show the patent right of the engine, breaks, wheelers etc., To solve this problem we propose two solutions.

The first solution is to use the search engines where the user can enter the queries and then he will get the related answers to that query. The second solution is to retrieve the relevant keywords. For this we use click-through data that is for any two queries that user enters we calculate the number of times the same result is retrieved then it is said to be relevant for both users and it is used for query expansion.

We can use both methods i.e., search engine and retrieving the relevant keywords for the users then we can select the keywords with highest clicks for query expansion.

3.5 Ranking

Ranking models are used to represent the patents that are having the highest search. Ay existing ranking function can be used in our search for patents and a good ranking model should consider the following factors:-

- The importance of patents is denoted by their weight and it is denoted by W_P
- We can denote the patents as a graph where patents are nodes and edges as connection between the patents.
- R (p,Q) denotes the keyword matching between the query entered by the user and the patent.
- T (p,Q) denotes the topic matching between the query entered by the user and the patent.
- The priority of the patent can be computed as follows:- $P(c/d_j)=(1-\eta)*count(c,d_j)/|d_j| +\eta *count(c,D)/|D|$
- The rank of the patent p with respect to the query Q is given by, $S(p/Q)= \alpha*W_P + \beta * R(p,Q) + \gamma * T(p,Q) + (1-\alpha-\beta-\gamma) * P_P$
- We use the above function to compute the matching between the patent and query submitted by the query and returns the top-K answers.

3.6 Patent Partition Selection

Once the user has entered the query it need not to be checked in all patent domains and it is checked in only particular domain and sub-domains. By this way we can improve the search efficiency of the users and also we need not check every patent domain we retrieve the top-k patent domains related to the query entered by the user and search only in those domains.

We need to know how to retrieve the top-k patent domains. For this purpose we need to consider the following:- First we need to check the topic-relevancy. It is nothing but checking whether the topics entered by the user and the topics in the patent domains are relevant. Second is the keyword relevancy. It is nothing but checking whether the keywords entered by the user and the keywords of the patents in the domains are relevant. We can use tf*idf model to calculate the relevancy. Third is the prior-art relevancy where we ensure that the patent domains consists of all the patents according their products. So, that the user can easily get the answers if they match with the patent domains. So by combining the above three factors we can find the top-k relevant patent domains.

3.7 Query Processing

As soon as the user has entered the query it will go for error-correction if there are any mistakes in the query then the topics related to the query are suggested then all the other products related to that query will be displayed then it will go for particular domain and sub-domain and the answer is returned to the user as quickly as possible. Meanwhile every patent visited will be given the priority based on this priority the rank of the patent will be increased.

IV. Experimental Study

We have taken patents from 2010 from USPTO data set and we have selected the top-k keywords from each partition where k ranges from 1 to 6 and these keywords are taken from the title and abstract then we compared them with the state-of-the-art method SVM_{PR}.

4.1 Quality Comparison

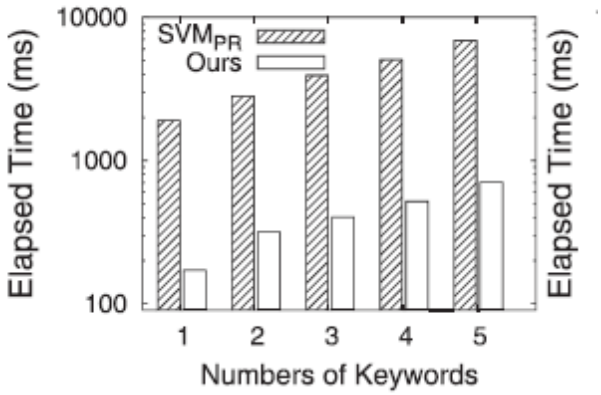
Here we represent a table that shows the experimental results. For Example:- p@20 error-correction improves the quality from 0.2 to 0.6 and topic-based query suggestion improves the quality from 0.8 to 0.10 and query expansion improves the quality from 0.7 to 0.14 and the overall by combining the three factors the quality can be increased to 0.88. this is obtained because of three techniques where the error-correction will automatically correct the wrong spelt keywords and the topic-based query suggestion will provide the topics related to the query entered by the user and query expansion is used to display all the other products related to that patent.

4.2 Efficiency Comparison

Here we first divide the patents into 24 domains and each domain consists of 3 sub-domains. Each domain is given an index. Then we check whether the efficiency of our search is improved or not but definitely it is improved as the patents are divided into several domains and sub-domains and the user when enters the query it is not searched in all domains and it will go the particular domain and sub-domains, by this way our search efficiency is improved.

V. Related Works

Larkey the author of "A Patent search and classification" has represented how to classify the patents but he neglected the fact of "prior-art search". Guo and Gomes the authors of "Ranking structured documents" proposed SVM patent ranking model to improve the search efficiency. Xue and Croft the authors of "Automatic query generation" has proposed methods to change the query entered by the user into search query and use the search query to find the relevant answers for the query entered by the user. They focused on how to extract the patents from the query entered by the user, how to rank them, but in addition to this we also find how to improve the efficiency and quality in extracting a patent based on the user entered query from the domains where the patents are classified.



VI. Conclusion

In our proposed system we developed three effective techniques error-correction, topic-based query suggestion, query expansion which makes the user easily search their required patents. Error-correction is used to correct the user's queries if there are any spelling mistakes or typing errors etc., Topic-based query suggestion is used to provide the users the topics for the users so that the users can easily search within their desired topic.

Query expansion is used to expand the query given by the user and displays all the related information to the user. Here we also partition the patents into groups so that when the user enters the query it will be

forwarded to the particular partition and the result will be quickly displayed to the user. Here, we also maintain separate databases for different patents. In future, Our proposed can be improved by making it more user-friendly to the user and by making it more efficient with high quality designing.