PatentQuest: A User-Oriented Tool for Integrated Patent Search

Manajit Chakraborty, David Zimmermann and Fabio Crestani

Faculty of Informatics, Università della Svizzera italiana, Lugano, Switzerland

Abstract

Patent Search is a well-established research field. In existing Patent Search systems, a user needs to explicitly enter a set of keywords to retrieve a set of ranked results. Conventional patent search systems lack the capability to run directly from the user's text editor. Moreover, to the best of our knowledge, most practical systems do not leverage explicit user feedback and domain-specific context to enhance the quality of search results. In this paper, we describe a system that offers a single point of access for patent information coming from different sources as well integrates the capability for user feedback and the ability to search from the text editor itself without the need to switch applications. To explore the viability and effectiveness of such a system, we created and deployed it as a web service plug-in for Microsoft Word®and conducted both system and user evaluation on a benchmark dataset.

Keywords

Patent Search, Integrated Search, Relevance Feedback, User Study, Add-in, Web Service

1. Introduction

In recent times the scale of intellectual property rights, including patents, have seen an unprecedented increment in the market globally. To keep up with this global phenomenon, patent offices in several countries are trying to improve patent prosecution quality while minimising the time required to grant patent rights without compromising the robustness of the patent evaluation structure. As such, the ownership of patents is fast becoming one of the most important measures of individual and business as well as national competitiveness. Hence, many companies have recently been encouraging and patenting the newest technologies in huge quantities. Compared to the increasing number of patent documents, the number of patent examiners and judges to handle them is not sufficient enough, and allocating the excessive workload to the limited workforce resources will inevitably deteriorate the quality of patent examination. Therefore, it is imperative for both the applicant and the examiner to perform the manual patent examination process both quicker and more accurate than before.

The patent search tasks have the following several purposes. One of them is 'prior art search', which has been required before patent filing or for the prevention of patent infringement. It is significantly different for the patent search system on that the purpose and characteristics

CEUR Workshop Proceedings (CEUR-WS.org)

of the existing search engines have long been endowed. While there are various public patent search systems like PatentScope[1], Espacenet[2] *etc.* and even commercial patent search systems such as Google Patents[3]. However, these systems often come with a big learning curve or are limited by their own data collections. Moreover, since patent prior-art search involves specific legalese, various firms offer patent search services like PatentSight[4] for a high price quote. For an inventor, especially a first-time patent applicant, it might seem both an overwhelming and expensive task. To address these issues, we demonstrate the viability of a more user-oriented system that is cost-free. This is achieved by direct integration into the user's text editor, allowing for search without reformulating the text into a query and working hand in hand with the user through a user feedback loop, leveraging domain-specific context information.

The stated goal is implemented with the construction of a functioning system prototype called *PatentQuest*. The system is deployed as a simple add-in to the online web service of Microsoft Word®. The advantage of such a system is that it incorporates the patent search within the text editor itself, thus allowing us to harness the power of explicit user feedback while allowing the user to access the patent text content, all without the need of switching between applications. The prototype system was evaluated on the CLEF-IP 2011 Prior-Art Search [1] track dataset for system performance and efficiency, while a separate user-study was conducted to gauge the system's usability and convenience.

2. Related Work

Patents pose several domain-specific challenges when it comes to information retrieval [2]. Further complications can arise from the fact that patents are written in different languages, are semi-structured and that the input for building a query can itself be a multi-page patent application [3]. To achieve better search performance, different techniques for query reformulation have been tested, and applied [4]. A potential technique for query expansion is "Pseudo-relevance feedback" (PRF), in which a first search based on an initial query is run and then features are extracted from the best scoring results to run a second run search. [5] produced an especially interesting result when they conducted their research on PRF. While they failed to demonstrate the better performance of their PRF techniques over the baseline keyword search, they found that the baseline performance can be doubled if just one extra document is marked as relevant by the user, suggesting that the interaction with the user is very powerful. Another approach to query expansion is the addition of synonyms or semantically related concepts to the given query terms. In the patent domain, different sources for the addition of this semantic information have been tested. Synonyms have been extracted from the general dictionary WordNet[5] or from the document corpus itself [6], domain-specific dictionaries have been built based on examiners' search queries [7] and Wikipedia[6] articles have been exploited for related sentences [8]. All of these systems mentioned above perform with rather mixed results.
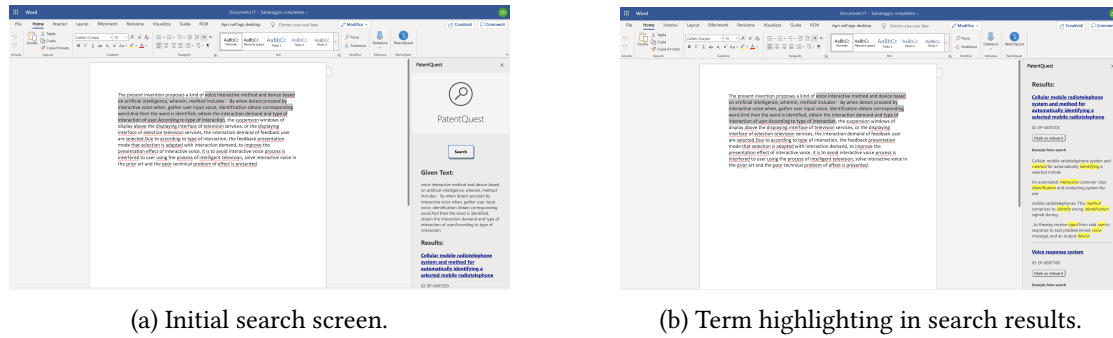
---

[1]https://www.wipo.int/patentscope/en/
[2]https://www.epo.org/searching-for-patents/technical/espacenet.html
[3]https://patents.google.com/
[4]https://www.lexisnexisip.com/products/
[5]https://wordnet.princeton.edu/
[6]www.wikipedia.org

(a) Initial search screen.

(b) Term highlighting in search results.

Figure 1: User interface with search results.

The IPC Classifications have been another source for query extension. Verma and Varma [9] built a classification vector for all patents in the data set and calculated the cosine similarity of the documents based on these vectors, which was the best performing system in CLEF-2011 Prior Art Search track. Patents also contain citation information that can be exploited in different ways. Mahdabi *et al.* [10] extracted citations from the text of the patent application and added those citations directly to the search results. Crestani *et al.* [11, 12] used the citations to build citation networks and exploit information gained from that network, among others using PageRank (see [13] for more details on PageRank).
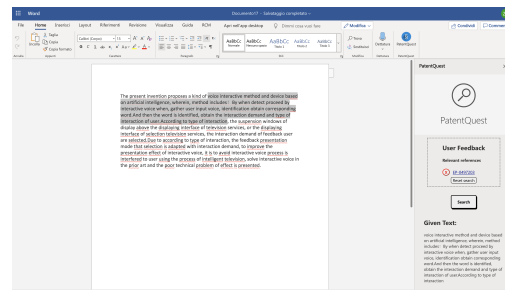
3. PatentQuest

In our literature review, we did not come across any system that offers the flexibility of Patent Search or Recommendation incorporated within a free-text document editor. In lieu of that, we built a prototype for our system called *PatentQuest* that facilitates users or inventors to have an integrated system at disposal that can fulfil their bibliographic needs while formulating a patent document. The prototype is distributed as an add-in to the online version of Microsoft Word® under Microsoft Office 365. Microsoft Word offers any developer to develop a piece of software or tool and integrate it with Word as an add-in without much hassle. This drove us to prepare a Patent Search prototype for Word. Although we intend to built an add-in for the desktop version of Microsoft Word in the immediate future, our goal here is to provide a proof-of-concept of such a prototype and the advantages it brings with it. In this section, we describe the user-interface and its characteristics.

The objective of the user interface is to keep the usage of the add-in as intuitive and self-explanatory as possible. After installation of the add-in, the user will see an additional icon under the "Home" tab. On clicking the icon, a side window opens for the user to interact with the system. The advantage of such a window-based system is that it allows the user to be free of unnecessary distraction while writing, as the window can be simply closed by re-clicking on the add-in button on the tab. A user is provided with basic instruction on how to use the system before starting running a search. The user is instructed to select a part of or the full text (in the current editor window) to run the search on and to click the *Search* button in the side window.

(a) Pop-up screen with patent details.

(b) Interface with user feedback information.

Figure 2: PatentQuest in use.

Queries can be issued in any of the three languages English, German or French. The search results show the English title of the patent, its document ID and an excerpt from the search, highlighting matching terms found from the query. By default, only the ten most relevant results are displayed with a link at the bottom of the window, which displays up to an additional 20 search results. The search results window is flexible, allowing the extended results to be hidden with a link at the bottom. A sample screenshot of the interface with search results is presented in Figure 1. For each search performed, the search query is preserved for reference purposes.

Clicking on the title of a patent opens up a pop-up window, displaying the content of a full document (Figure 2a). The editing space and search results are static in the background while the full-text of the patent is displayed in the scrollable pop-up window. This allows the user to have a comfortable reading experience without losing the search results or the text written so far in the editor window. In the display window, the upper section is devoted to meta-data information *i.e.,* the title of the patent in English, and the document ID is shown at the top followed by the file types (whether it is an application or grant). In the lower section, the rest of the relevant sections extracted from the patent document are displayed, like the citations, the abstract, the claims or the description.

Additionally, each search result comes with a button to mark it as relevant (Figure 2b). Once a document has been marked as relevant, a panel shows up on top of the search button displaying a list of documents (sorted by their document ID) that have been selected by the user. The panel also allows to delete the documents again or display them in full by clicking on the ID. The user has the option to issue a new search at any time. If given documents are marked as relevant, they will be taken into account for the new search (see section 5.1) for an explanation on relevance feedback). Once the user has received is satisfied with the desired search results, the side window can be closed again by clicking on the cross button on the top right corner. The side window can be re-opened by clicking on the add-in's panel in the "Home" tab. As long as the application window is not closed and reloaded, the current session's search results and documents marked as relevant remain intact. The option to select and mark relevant documents from search results as relevant (explicit relevance feedback) helps user to drive their navigation in a specific direction and has been shown to improve prior-art search previously [14]. This is particularly helpful if an inventor is looking for similar or seminal patents on a specific topic.
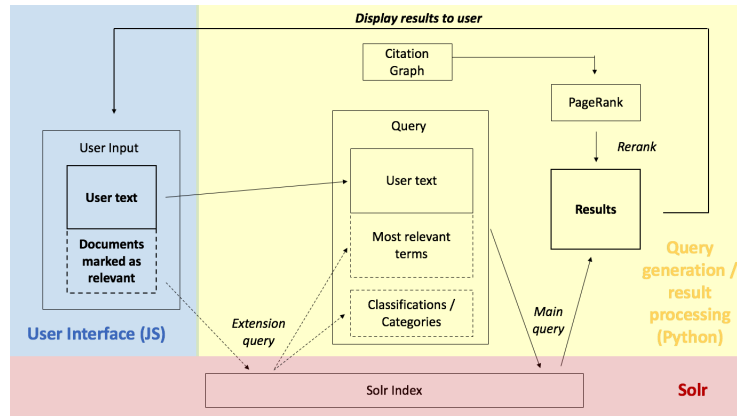
Figure 3: Schematic view of system implementation and data flow.

4. Implementation Details

4.1. Dataset

We used the dataset from CLEF-IP 2011 track[7] for building and evaluation of our system. The dataset consists of over 1 million patents from the European Patent Office[8] prior to the year 2002 and additional 400,000 patent applications published by the World Intellectual Property Organization in XML format.[9] The elements of the XML files can be roughly divided into two categories: Text fields with the contents of the patent and fields with meta information. The text fields of a patent are: *Title, Abstract, Description* and *Claims.* In the dataset, the text elements of a patent can be in one of three languages: English, German or French. Generally, the title of the patent is available in all three languages and other text fields in only one language. For each patent, several documents can be published, depending on the information available at the time of publishing. The documents are encoded with "A1", "A2", ... for the application phase and with "B1", "B2", ... for the granting phase. The relevant information about the patents is spread over several documents in some cases. Overall the dataset comprises of around 2.5 million files.

4.2. System Design

The system design can be broadly divided into two parts: (i) the front-end or user-inteface (ii) the back-end handling the query processing and display of the results. In Figure 3, we present an overview of the implementation of the system.

4.2.1. User Interface

PatentQuest was motivated by the lack of an integrated system within existing workflows. The system thus built should complement the creation of documents by always suggesting relevant

---

[7]http://www.ifs.tuwien.ac.at/~clef-ip/download/2011/index.shtml
[8]https://www.epo.org/
[9]www.wipo.int

sources based on the current text written by the user. This is beneficial since it saves the user both the time and effort required to switch between applications to gather relevant sources to cite. An ideal system in this scenario should provide the user all the functionalities of a standard text editor while also providing not only an integrated list of relevant search results but one that also allows adding parts of a relevant source directly into the text. In addition, the system should also be backwards compatible with existing documents. All of this suggests that the best way would be to apply one of the more popular text editor currently in use by users and experts as a user interface that offers the possibility of customisation. In light of this, Microsoft Word stands out as a favourable choice for this task. Microsoft offers two ways to create extensions to their "Office Suite" programs, both with their own caveats: (a) Office Add-ins and (b) COM/VSTO Add-ins.

As mentioned earlier, our system is built as an Office Add-in. This is the newer format to create an add-in for Microsoft Office products. All office applications offer a JavaScipt API ("Office JS") to access the contents of the document and offer a browser engine that runs in a side-window of the application to render HTML5 and CSS, as well as execute JavaScript. This form of the add-in is cross-platform compatible, unlike the COM/VSTO add-in, which was the main reason it was chosen for the implementation of the prototype system. It also offers strong security through the limited access of the JavaScript add-in on the user's system. At the same time, the strong security measures implemented by Microsoft induce the biggest drawbacks of this form of the add-in. It forces the add-in to be run as a web service. The Office application will only load an add-in that is served through the HTTPS protocol, making a local standalone use of the add-in difficult. Furthermore, the distribution of the manifest file, which contains the necessary loading information for the add-in, is built for either distribution through an organisation with a central IT infrastructure or through the Microsoft AppSource, which requires authorisation from Microsoft. While this limits our system by allowing us to use only the online version of Microsoft Word through the Office 365 suite, it still allows enough provision for both system and user evaluation. We aim to offer a standalone add-in in the near future. For our current prototype, the user needs to side-load the add-in through the distribution of a manifesto file. The user has to simply download an XML file including the manifesto with the required information and select it through a file manager to integrate the add-in.

4.2.2. The back-end

There are several reasons that compelled us to deploy the system as a web service. Firstly, the chosen form of Microsoft Office add-in requires a connection to a secure web service based on the HTTPS protocol to work. Secondly, distributing the full Solr Index (explained in the next section) to all end users would be very heavy (around 40GB) and ensuring the timeliness of updates for the dataset becomes much easier. Finally, setting up the system becomes extremely easy for the user. All the user has to do is load the manifest XML file into his Microsoft Word online distribution. Flask[10], a web application framework for Python, was chosen as a foundation for the implementation of the back-end and to connect the APIs needed by the front-end to

---

[10]https://flask.palletsprojects.com/en/1.1.x/

the web. The more advanced features of the system raise the need for query generation or re-formulation and refreshing the results on-the-fly. When a search request is triggered, the user interface sends two types of information to the Flask app in the back-end, the text selected for query input and the patent IDs of documents marked as relevant by the user (relevance feed-back). The user input text is used as the original query, which is then expanded in two stages using the documents marked as relevant:

- by adding the most important terms from the selected documents and then

- by adding the IPC classifications to the original query.

The relative weights involved in the query expansion and reformulation have been determined empirically (see section 5.1). The important terms are extracted using the "More Like This" MLT feature of Apache Solr[11]. In the same stage, the categories/IPC classifications of the documents are collected. This information is then used to reformulate the user request and build the final query, which is again run against the search index.

5. Evaluation

In this section, we describe the evaluation process and the evaluation results of the system. The aim of the evaluation is to measure the quality of the search results as well as the usability of the system and to gain insights into potential improvements. Hence, we conducted both system evaluation and user evaluation to gain a fair understanding of the strengths and limitations of our system. For the system evaluation, the CLEF-IP 2011 patent dataset is used to determine the optimal system parameters and to compare the system's performance with that of the systems participating in the CLEF-IP 2011 track. This is followed by the evaluation of the system by a test user group.

5.1. System Evaluation

In CLEF-IP 2011, the participants were provided with 3,973 topics in three languages (English, German and French). Various textual and non-textual elements (as described in later sections) were also employed by the winning systems of the CLEF-IP 2011 track and were the starting point for the queries tested below. We conducted several experiments to determine the opti-mal settings and weights for each parameter used in the design of the system to have the best possible performance. Each of these empirical studies is presented below. The evaluation met-rics used in all these experiments were (i) Mean Average Precision (MAP) and (ii) Normalized Discounted Cumulative Gain (nDCG), which were the metrics used to judge the participant teams' performance in the CLEF-IP 2011 Prior-Art Search track.

5.1.1. Impact of combining different patent sections

As described in Section 4.1, a patent document consists of multiple sections. The first experi-ment thus compares the search result metrics for queries generated from different combinations
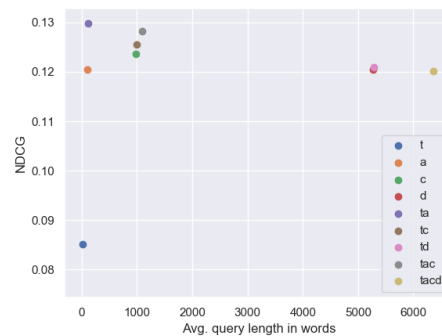
---

[11]https://lucene.apache.org/solr/guide/6_6/morelikethis.html

Table 1
Determining optimal combination of patent sections

Section	MAP	nDCG	Avg. query length (#words)
t	0.0394	0.0851	9.39
a	0.0585	0.1205	106.09
c	0.0612	0.1236	978.95
d	0.0620	0.1205	5270.01
ta	0.0633	**0.1298**	114.48
tc	0.0623	0.1255	987.34
td	0.0622	0.121	5278.40
tac	**0.0635**	0.1282	1092.43
tacd	0.0615	0.1202	6361.44
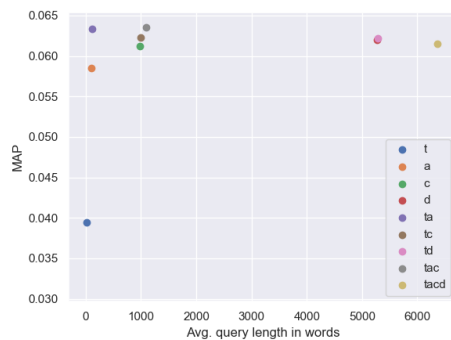
(a) MAP vs. Query Length.   (b) nDCG vs. Query Length.

Figure 4: Determining the optimal combination of patent sections.

of sections, which in turn translates to varying query length. The query configurations are represented by an encoding where a combination of letters describes the combination of the fields used. The encodings can be described as *t*: Title, *a*: Abstract, *c*: Claims, *d*: Description, *ta*: Title + Abstract, *tc*: Title + Claims, *tac*: Title + Abstract + Claims and *tacd*: Title + Abstract + Claims + Description. Figures 4a and 4b shows the impact on retrieval performance for choice and combinations of different sections from the patent.

As can be seen for the queries "t" to "d", there is an increase in MAP for using longer text elements like description and claims versus title or abstract but with rapidly declining marginal returns and at the expense of longer run times. When the titles are combined with the abstract or the claims, the advantage of using the descriptions subsides, despite the titles only adding an average of around nine words (the descriptions have an average length of around 5,270 words). Combining the description with a query that already contains the titles and the abstract and/or the claims seems to add noise rather than any useful information. This result is confirmed by the run, which combines all the elements, having lower performance on all metrics. The tendency for longer queries to do better in the prior art search task at the expense of query
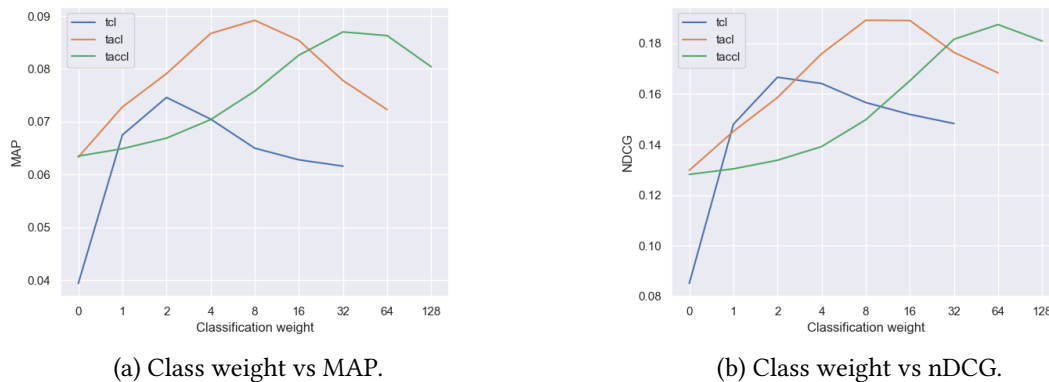
(a) Class weight vs MAP.

(b) Class weight vs nDCG.

Figure 5: Effect of adding Classification Codes on performance.

speed confirms the results of [15]. As can be seen from Figures 4a and 4b, two combinations show the best search performance, while having reasonable query length. The first one is title and abstract, and the second one is title, abstract and claims. Hence, we proceed with these two combinations for the rest of the experimentation.

5.1.2. Impact of incorporating classification codes

As mentioned earlier, each patent document and the topics have their constituent classification codes, which we used to improve the retrieval performance by adding the IPC codes to the optimal query generated from previous step. Figure 5a and 5b show the results for different boost factors (weights) for the classification codes of the two best queries from the previous experiment ("ta" and "tac") as well as for the title only ("t") query. We can observe that incorporating IPC classification within the query adds considerable information that had not been previously captured by the text alone. Secondly, one needs to adapt the boosting in accordance with the query length. For instance, for a query built from the title and abstract ("ta"; average length of 114.5 words) of the patent application, the optimal weight of the classifications is eight times the weight of the terms ("tacl_8"), while for a query consisting of title, abstract and claims ("tac"; average query length of 1092.4), the classifications should be assigned 32 times the weight of each term ("taccl_32"). Interestingly, after taking into account the classification codes, terms from the claims section do no longer seem to add useful information compared to just using title and abstract.

### 5.1.3. Impact on multi-lingual search

Next, we studied the effect of retrieval performance due to languages. As stated earlier, in the dataset, we had documents from three languages, namely, English (en), German (de) and French (fr). Table 2 present the results obtained.

One can observe that the results differ significantly, depending on the input languages. We observed that queries in English produced the best results, closely followed by German, while system struggled the most with queries in French. However, not only are the inputs given in

Table 2
Effect of multi-lingual search on performance.

Query type	MAP		nDCG	
	Single lang.	Multi lang.	Single lang.	Multi lang.
tcl_2	0.0746	0.0735	0.1666	0.1653
tacl_8	0.0892	0.0886	0.1891	0.1959
taccl_32	0.0870	0.0913	0.1816	0.1928
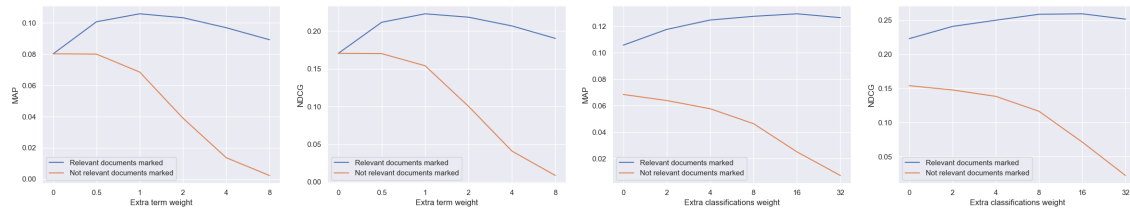
(a) Term weight vs. PRF (MAP). (b) Term weight vs. PRF (nDCG). (c) Class weight vs. PRF (MAP). (d) Class weight vs. PRF (nDCG).

Figure 6: Effect of Pseudo-Relevance Feedback.

three languages but also cross-language results are expected. An input given in German might expect a result published in French and the other way round. Since the best faring query uses only the relatively short abstract and the title, an attempt was made to use machine translation to achieve better results. The titles themselves are usually already given in all three languages. For each input patent, machine translations of the abstract were created for the other two missing languages, using the Yandex translation API[12]. Employing a multi-lingual search, the performance of the system improved in most cases, and the best results were achieved when the combined query of title, abstract and classification weight of 32 ("tacl_32") was used.

5.1.4. Impact of Relevance Feedback

As part of our next experiment, we wanted to determine if relevance feedback could improve the system performance even further. For this, we employed pseudo-relevance feedback (PRF) in two ways: (a) by selecting top-2 relevant results returned by the optimised query and expanding it and (b) by selecting top-2 non-relevant results for query expansion. This experiment helped us realise two objectives, (i) whether our system was indeed responsive to relevance feedback in the first place and (ii) the optimal weight to be considered for the same. Figures 6a and 6c, compare the MAP performance against the term weight and classification code weight boosting, while figures 6b and 6d, compare the nDCG performance. In both cases, we can clearly observe that positive relevance feedback can improve retrieval performance considerably. In fact, the best-run results obtained by our system after positive relevance feedback on *tacl_32* was 0.0905 in terms of MAP and 0.205 in terms of nDCG, which were comparable with

---

[12]https://tech.yandex.com/translate/

Table 3
Summary statistics SUS score for user group evaluation.

	Mean	Max.	Min.	Std. Dev.
SUS	72.5	92.5	50	18.14

the best performing system at CLEF-2011 PAC track (MAP=0.097 [9]). While better results have been achieved in the meantime (*e.g.* by Mahdabi and Crestani [12]), it needs to be noted that our objective was not to provide a mechanism for best search results but to achieve a balance between system performance and usability of the system to general users while providing a novel integration[13].

5.2. User Evaluation

While system evaluation could provide us with a measurable impact on the system performance, since the tool is designed for users, it was imperative that we conducted a user evaluation as well. In the absence of expert users, we resorted to a set of four users with high familiarity with IR systems. To evaluate the usability of the system, the "System Usability Scale" (SUS) was employed [16]. The SUS score as recorded for the four test users is presented in Table 3. From the table, we can observe that the standard deviation of the SUS is 18.14, which implies that there is a wide range of different perceptions of the system (between 50 and 92.5). When translated to the various scales of SUS evaluation [17], the SUS of 72.5 corresponds to a grade "C" or a qualitative description of "Good". This lies within the range of what users tend to deem acceptable. Naturally, smaller improvements could probably yield substantial improvements in the scoring. Along with the SUS evaluation, the test users were asked to record their responses to an additional questionnaire. The questionnaire has two blocks of statements. The first block contains five extra statements about usability, which were more specific to the system than the SUS statements. The extra statements were only presented to the participants after they had completed rating the SUS statements in order not to influence or bias the questionnaire. The second block of extra statements consists of three statements about the subjective quality of the search results. The summary statistics recorded against the questionnaire is presented in Table 4. At the end of the questionnaire, the test users were presented with two text fields to add general feedback and overall suggestions for improvements to the system. From Table 4, we can observe that on the usability statements, the system was again perceived very differently by different users. The user group's scoring confirms that the response time and the inclusion of the search into the natural workflow of document creation belong to the strengths of the system. On the other hand, the aesthetics of the interface received a score below average, indicating room for improvement, like changing the colouring or hiding the button (for relevance feedback) after a document has been marked relevant, to have a more polished user-interface outlook. However, the overall user study substantiated our initial goal of building a prototype system capable of integrating patent search within the document editor freeing the user from having to switch between workspaces. We have duly recorded the feedbacks and suggestions

---

[13]http://www.dlib.org/dlib/november95/11croft.html

Table 4
Summary statistics of user group scoring of the additional statements.

Statement	Mean Score	Std. Dev.
The add-in is easy to install.	3.25	1.71
The system's response time to search requests is adequate.	3.75	1.89
The user interface has an appealing look and feel.	2.75	1.50
The system can enhance the workflow of creating a new patent document.	3.75	1.26
The system is responsive to the user feedback loop.	3.25	1.26
All search results shown are relevant.	3.25	0.5
The most relevant results show up on top.	3	0.82
The search brings up all documents that are relevant to the search.	3.25	0.96

provided by the test users and intend to incorporate them in the next version of our system.

6. Conclusions and Future Work

Patent Search continues to be an active research area. In this paper, we demonstrated a prototype system that could allow integration of patent search within a popular text editor. The prototype deployed as a Microsoft Word add-in facilitates hassle-free integration into the text editor window freeing the user from the need to switch between applications for prior-art search. The system also allows user to provide relevance feedback to allow for a precision-oriented search while also providing the added advantage of handling multiple languages. We tested and evaluated our system, using a standard benchmarking dataset, from both the efficiency and usability perspectives. We showed that adding domain-specific information like IPC classification code, along with machine-translated text contents for multi-lingual search, improved the system performance. While the impact of explicit relevance feedback could not be determined quantitatively, we showed with the help of pseudo-relevance feedback that our system responded positively in the presence of correct relevant results. Moreover, the overall usability of the system was received quite favourably by the test user group.

While the prototype system was well-received overall, there are further potential improvements to the design which needs to be explored. Firstly, we would like to build an add-in for the use of a standalone local Word installation. Secondly, we aim to achieve better system performance by incorporating the lexical and semantic features of a patent document to account for the several unique factors of a patent, such as *obfuscation*. Finally, although the PageRank experiment (not discussed in the paper, for brevity) performed poorly in our case, we would continue to investigate and improve the integration of such network flow metrics to better system performance. Finally, we plan to incorporate all the additional suggestions by the test users to improve the user-interface even further to provide it with a more polished outlook.

References

[1] F. Piroi, M. Lupu, A. Hanbury, V. Zenz, Clef-ip 2011: Retrieval in the intellectual property domain, 2011.

[2] M. Lupu, K. Mayer, J. Tait, A. Trippe, Current Challenges in Patent Information Retrieval, volume 37, 2017.

[3] W. Shalaby, W. Zadrozny, Patent retrieval: A literature review, Knowledge and Information Systems (2019) 631–660.

[4] G. Cabanac, I. Frommholz, P. Mayr, Bibliometric-enhanced information retrieval 10th anniversary workshop edition, in: European Conference on Information Retrieval, Springer, 2020, pp. 641–647.

[5] M. Golestan Far, S. Sanner, M. R. Bouadjenek, G. Ferraro, D. Hawking, On term selection techniques for patent prior art search, 2015.

[6] W. Magdy, G. Jones, A study on query expansion methods for patent retrieval, International Conference on Information and Knowledge Management, Proceedings (2011).

[7] W. Tannebaum, P. Mahdabi, A. Rauber, Effect of log-based query term expansion on retrieval effectiveness in patent searching, in: Experimental IR Meets Multilinguality, Multimodality, and Interaction, Springer International Publishing, Cham, 2015, pp. 300–305.

[8] B. Al-Shboul, S.-H. Myaeng, Query phrase expansion using wikipedia in patent class search, in: M. V. M. Salem, K. Shaalan, F. Oroumchian, A. Shakery, H. Khelalfa (Eds.), Information Retrieval Technology, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011, pp. 115–126.

[9] M. Verma, V. Varma, Exploring keyphrase extraction and ipc classification vectors for prior art search., in: CLEF (Notebook Papers/Labs/Workshop), 2011.

[10] P. Mahdabi, L. Andersson, A. Hanbury, F. Crestani, Report on the clef-ip 2011 experiments: Exploring patent summarization, volume 1177, 2011.

[11] P. Mahdabi, F. Crestani, The effect of citation analysis on query expansion for patent retrieval, Information Retrieval 17 (2013) 412–429.

[12] P. Mahdabi, F. Crestani, Query-driven mining of citation networks for patent citation retrieval and recommendation, CIKM 2014 - Proceedings of the 2014 ACM International Conference on Information and Knowledge Management (2014) 1659–1668.

[13] S. Brin, L. Page, The anatomy of a large-scale hypertextual web search engine, Computer Networks and ISDN Systems 30 (1998) 107 – 117. Proceedings of the Seventh International World Wide Web Conference.

[14] S. Bashir, A. Rauber, Improving retrievability of patents in prior-art search, in: Advances in Information Retrieval, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010, pp. 457–470.

[15] D. Becks, M. Eibl, J. Jürgens, J. Kürsten, T. Wilhelm-Stein, C. Womser-Hacker, Does patent ir profit from linguistics or maximum query length?, volume 1177, 2011.

[16] J. Brooke, "SUS-A quick and dirty usability scale." Usability evaluation in industry, CRC Press, 1996. ISBN: 9780748404605.

[17] A. Bangor, P. T. Kortum, J. T. Miller, Determining what individual sus scores mean: adding an adjective rating scale, Journal of Usability Studies archive 4 (2009) 114–123.