Automatic IPC encoding and novelty tracking for effective patent mining

Douglas Teodoro, Emilie Pasche, Dina Vishnyakova, Christian Lovis
Bibliomics and Text Mining Group
Medical Informatics Service
University of Geneva
4 rue Gabrielle-Perret-Gentil
1211 Geneva, Switzerland
{douglas.teodoro, emilie.pasche, dina.vishnyakova, christian.lovis}@unige.ch

Julien Gobeill, Patrick Ruch
Bibliomics and Text Mining Group
Library and Information Sciences Department
University of Applied Sciences
7 route de Drize
1227 Carouge, Switzerland
{julien.gobeill, patrick.ruch}@hesge.ch

ABSTRACT

Accurate classification of patent documents according to the IPC system is vital for the interoperability between different patent offices and for the prior art search task involved in a patent application procedure. It is essential for companies and governments to track changes in technology in order to asses their investments and create new branches of novel solutions. In this paper, we present our experiments from the NTCIR-8 challenge to automate paper abstract classification into the IPC taxonomy and to create a technical trend map from it. We apply the k-NN algorithm in the classification process and manipulate the rank of the nearest neighbours to enhance our results. The technical trend map is created by detecting technologies and their effects passages in paper and patent abstracts. A CRF-based system enriched with handcrafted rules is used to detect *technology*, *effect*, *attribute* and *value* phrases in the abstracts. Our experiments use multi patent databases for training the system and paper abstracts as well as patent applications for testing purposes, thus characterising a cross database and cross genre task. In the subtask of *Research Papers Classification*, we achieve a MAP of 0.68, 0.50 and 0.30 for the English and 0.71, 0.50 and 0.30 for the J2E subclass, main group and subgroup classifiers respectively. In the *Technical Trend Map Creation* subtask, we achieve an F-score of 0.138 when detecting technology/effect elements in patent abstracts and 0.141 in paper abstracts. Our methodology provides competitive results for the state of the art, with the majority of our official runs being ranked within the top two for both trend map (papers) and IPC coding. That said we see room for improvements especially in the detection of technologies and attributes elements in abstracts. Finally, we believe that the subtask of *Technical Trend Map Creation* needs to be adjusted in order to better produce a patent map. The classification system is available online at http://pingu.unige.ch:8080/IPCCat.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Applications and Expert Systems; H.3.3 [**Information systems**]: Information Search and Retrieval

General Terms

Algorithms, Performance, Experimentation

Keywords

IPC encoding, technology tracking, k-NN, CRF

1. INTRODUCTION

The four most relevant official patent classification systems are the European Patent Classification (ECLA), the Japanese File Index and File-forming Terms (FI/F-terms) classification, the US Patent Classification (USPC) and the International Patent Classification (IPC). However, only the IPC is used in a worldwide context, having 95% of all patent documents classified according to it and used in more than 100 countries. As such it is a very powerful tool for searching different patent-related databases, unlike ECLA, FI/F-terms or USPC, which are restricted to their respective offices. IPC is a taxonomy developed and managed by the World Intellectual Property Organisation (WIPO). The current version (8th) has been available since 2006 in English and French and contains approximately 70000 codes.

Automating the attribution of IPC codes to patent applications is important for several reasons: it assists patent experts in the patent classification task, aids inventors with the prior art search and helps referees to validate or refute a given application. Moreover, the use of the assigned IPC code is key information for searching patents across nations. Furthermore, when a patent application is considered or submitted, the search for previous inventions in the field - known as prior art - relies crucially on accurate patent classification.

Due to the increased rate of technological development and to a tougher economic scenario, it is important to companies and governments to understand how technology fields evolve over time. Comprehending the path and the effects of a given technology helps stakeholders to direct R&D investments and plan new products. A simple example that illustrates how important it is for companies to know the tendencies in the market, is a company that in the last decade worked with analog photograph cameras and had to change quickly to digital, losing and gaining market share depending on how fast they realised the new trend. Moreover, technological mapping is an important step in the generation of new inventions.

In this paper, we report the experience of the BiTeM group[1] in the Patent Mining task of the NTCIR-8 workshop [19]. The task is subdivided into two subtasks: 1) *Research Papers Classification* (RPC) and 2) *Technical Trend Map Creation* (TTMC). In subtask 1, paper abstracts are automatically encoded using the IPC categories. Our team joined two sub-subtasks of subtask 1: English, which consists of classifying English paper abstracts using English patent databases and J2E, where Japanese paper abstracts are classified using English patent collections. In subtask 2, technology and effect passages in paper and patent abstracts are tagged. We participated in the English sub-subtask .

We use two patent collections and a set of manually tagged abstracts to train our system. The assessment of our approaches are performed using 633 English and 639 Japanese paper abstracts in the *RPC* subtask and 200 paper and 200 patent abstracts in the *TTMC*. In order to improve classification we develop several re-ranking techniques which are further described. For the *TTMC*, handcrafted rules are created to help in the named entity recognition task.

The rest of this paper is organised as follows. In Section 2, the state of the art in automatic patent classification and patent mapping are introduced. In Section 3, the corpora and training data are depicted. Furthermore, we describe the k-NN re-ranking techniques and the methods used to detect technology and effect phrases in abstracts. In Section 4, the results obtained are presented. In Section 5, concluding remarks are discussed.

2. BACKGROUND

Due to delay in patent application analysis caused by the growth of applications, new areas of knowledge, size of patent databases and of the IPC taxonomy, automatic patent classification has become a hot topic for research in the last few decades.

Chakrabarti et al. [1, 2] implemented a hierarchical Bayesian classification system. In their system, between 5 and 10% of the vocabulary is used to distinguish documents at each level of the taxonomy. The hierarchical scheme is claimed to be faster and to increase slightly the system's accuracy. Larkey [15] presented a web-based system for the retrieval and classification of patent text implemented for the USPTO. The classification part of the system is based on the k-NN algorithm. Unfortunately, the authors do not present any results of the system's accuracy. Krier and Zaccà [13] describe a challenge carried out by several companies and institutions to classify applications into the ECLA taxonomy. From their tests, it turned out that the classification precision requested by the EPO, 81% at the directorate level at 100% recall, was too high. For 100% recall, the best reached precisions at the cluster, directorate, subclass and team levels were 80%, 72%, 61% and 57% respectively.

These methods could not be compared due to the lack of benchmarks and a freely available collection. In 2003 with the change of IPC version 7th to 8th, Fall et al. [3] developed a new collection, WIPO-alpha, and proposed a new evaluation method for patent documents (TOP, ALL and 3 GUESSES). Since then, some authors [26, 27] have reported using this evaluation method in addition to the classical precision/recall performances. In the same work,

[1]http://eagl.unige.ch/bitem

Table 1: Code's distribution for the PAJ and USPTO corpora.

# \ depth	corpus	class	sub-class	main group	sub-group
codes	PAJ	104	420	4738	30885
	USPTO	104	428	6588	38491
average codes/doc	PAJ	1.4	1.5	1.9	2.3
	USPTO	1	1	1	1
max docs/code	PAJ	333413	180552	123062	24364
	USPTO	83409	50103	17880	5026
min docs/code	PAJ	54	13	1	1
	USPTO	19	1	1	1
average docs/code	PAJ	22910	5673	503	77
	USPTO	8549	5602	135	23
median docs/code	PAJ	12737	3497	181	35
	USPTO	2722	706	14	5

the authors reported their results of applying a variety of machine learning algorithms to automate categorisation of English-language patent documents into the IPC. Their results show that SVM outperforms the Naïve Bayes, k-NN and SNoW approaches under similar conditions, particularly for categorisation at the IPC subclass level.

Recently, Trappey et al. [28] developed a classification system based on neural network technology. Their system extracts key phrases from the corpus by means of automatic text processing and determines the significance of key phrases according to their frequency in the document set. Then, it applies the back-propagation network model in the classification process. The results yielded are above 90% of accuracy at the subclass level. The system is limited by the availability of computational resources but also, their training set is restricted, covering a small part of the IPC.

Xiao et al. [32] reported their experience in the NTCIR-7 workshop to classify paper abstracts according to the IPC taxonomy. Their approach is based on the k-NN framework, in which various similarity calculation and ranking methods are used. They achieved 48.86% of performance (MAP) when classifying according to the subgroup level. Additionally to their report, the workshop produced a series of other valuable works [18, 31, 4, 24].

With the need for new solutions, keeping track of technologies already developed is crucial for companies. An efficient way to find out how a given technology field evolves, its effects and how it interacts with other technologies is by means of a patent map. Recently, the subject has been increasingly explored in the literature and most of the work published is based either on keyword clustering [16, 11] or on citation networks [30, 7, 8, 9].

No and Park [20] analyse how technology evolves using citation network analysis. They are particularly interested in technology fusions. They propose a method to visualize the relationship between patents and their backward and forward patent citations, at the patent class level, with their direction on a citation map. Zhua et al. [36] use keywords but also authorship and noun phrases to create the patent map. Their system is based on Latent Semantic Indexing (LSI) to extract the main relationships implicit in a data set. Kim et al. [10] collet keywords from patent documents of a target technology field and cluster patent documents by the k-Means algorithm. With the results, a semantic network of keywords is formed. Then, the patent map is constructed by rearranging each keyword node of the semantic network according to its earliest filing date and frequency in patent documents.

These ways of creating patent maps show the general technology shift but do not necessarily highlight a specific technology snapshot. As far as we are concerned, there are no previous reports on creating maps at the technology level as proposed by the NTCIR-8 challenge, identifying technology and its effects terms in documents. An introductory work is presented by Tsend et al. [29], however their work needs to be further investigated. They propose that by clustering the problems and solutions individually a technology-effect map can be created. Tagging at the sentence level can be found in the works of Mizuta and Ruch [17, 22]. Especially in the bioinformatics literature, tagging at the term level, also called Named Entity Recognition (NER) or Sequence Tagging, is reported by several authors [25, 12]. Finally, we can cite a patent document which describes a part of speech tagging system for content-word pairs [35].

3. METHODS AND DATA

3.1 Training data

In our experiments for the *RPC* subtask, we use the Patent Abstracts of Japan (PAJ) (2382595 documents) and USPTO (889116 documents) collections[2] to train our classifiers. The number of documents versus code distribution for the training corpora are displayed in Table 1. We use only the TITLE and ABSTRACT tags for indexing the PAJ collection. The USPTO corpus contains other interesting fields such as CLAIM, SPECIFICATION and CITATION, additionally to the TITLE and ABSTRACT tags presented in the PAJ collection. USPTO also differs from PAJ in the number of codes assigned to documents: PAJ contains main and secondary codes while USPTO contains only the main IPC code. For this collection, we create two different indexes: USPTO and USPTO_CLAIM. The latter contains TITLE, ABSTRACT and CLAIM tags while the former contains, as the PAJ collection, only TITLE and ABSTRACT.

The organisers also provide two sets of 976 paper abstracts each, which have IPC codes manually assigned by experts. One set contains English paper abstracts while the other, Japanese abstracts. From the way the task is designed, it characterises a cross-genre and cross-database (and cross-language for J2E subtask) classification, where the domain of queries (paper abstracts) differs from that of the retrieval target (patent) and the application (a research paper abstract without any office) differs from the office patent database(USPTO and JPO).

For the *TTMC* subtask, the organisers provide two sets of 300 abstracts each, which have *technology* and *effect* passages tagged. The *effect* tag contains further *attribute* and *value* elements annotated. One set is composed by paper abstracts while the other by patent abstracts. Fig. 1 shows an example of a tagged abstract. A more detailed description of the training and test collection can be found in [19].

[2]We do not use all the documents from the PAJ and USPTO collections, which is approximately 3/2 bigger than the sample we use for training. Instead, we use only documents which had the IPC codes already provided in a separate mapping file.

Figure 1: A sample of a tagged abstract.
<ABSTRACT>In this paper, a double layered self-diplexing antenna (SDA) using <TECHNOLOGY>circular microstrip antenna</TECHNOLOGY> for transmitting and ring patch antenna for receiving has been analysed by a cavity model to clarify the mechanism of the mutual coupling between the transmitting and the receiving antennas. Key factors to <EFFECT><VALUE>reduce</VALUE> <ATTRIBUTE>the mutual coupling</ATTRIBUTE> </EFFECT> are the arrangement of the feed pin location for 2-feed SDA and <EFFECT><VALUE>minimizing</VALUE> <ATTRIBUTE>the amplitude error</ATTRIBUTE> </EFFECT> of hybrid circuits for 4-feed SDA.We also verified analysis results by experiments.</ABSTRACT>

3.2 Classifier design

We use Terrier[3] as our information retrieval (IR) engine. Terrier implements several methods to calculate the similarity between documents: BM25, BB2 (Bose-Einstein model for randomness), InL2 (inverse document frequency model for randomness), among others and it is optimised to work with large collections. It is based on JAVA and freely available online.

For our classification experiments, we choose a classifier based on the k-NN algorithm. It is an empirical decision derived from other studies that show that k-NN, together with SVM, outperforms other approaches such as neural networks, Rocchio and Naïve Bayes [23]. Compared to SVM, k-NN scales much better to larger systems that contain many features and classes, which is the case in the proposed task.

Weighted k-NN classifiers have been consistently strong performers in text categorization evaluations [33, 34]. The $tf \cdot idf$ weight (term frequency - inverse document frequency) model is a statistical technique used to evaluate how important a given word is to a document. In this model, the importance of a term t is directly proportional to the number of times t appears in a document d, and is inversely proportional to the frequency of the documents, in which t is contained, in the corpus c. Thus, a high weight in $tf \cdot idf$ occurs when t has a high frequency in one document and a low document frequency in the whole collection of documents. This causes common terms in the collection to receive low weights. Terrier uses the Okapi BM25 $tf \cdot idf$ formula to perform the document ranking, where the idf factor $w^{(1)}$ (see [21]) is normalised as follows:

$$w(t,d) = w^{(1)} \frac{(k_1+1)\,tf}{K+tf} \frac{(k_3+1)\,qtf}{k_3+qtf}, \tag{1}$$

where $w(t,d)$ is the weight of document d for query term t. The sum of $w(t,d)$ of the query terms gives the final weight of document d. K is given by

$$K = k_1 \left((1-b) + b\frac{l}{l_{avg}} \right), \tag{2}$$

where l is the document length and l_{avg} is the average document length of the collection. The query term frequency qtf is the number of occurrences of a given term in the query and tf is the within document term frequency of the given

[3]http://ir.dcs.gla.ac.uk/terrier

Table 2: k's for the different classifiers.

corpus	subclass	main group	subgroup
PAJ	16	17	11
USPTO	158	122	130

term. The constant b is the free parameter of the BM25's term frequency normalisation method, which can be seen as:

$$tfn = \frac{tf}{(1-b) + b\frac{l}{l_{avg}}}, \tag{3}$$

where tfn is the normalised term frequency. The parameters k_1, k_3 and b are tuning constants. k_1 and k_3 changes the influence of term frequency on the document and query respectively. Higher values of k_1 or k_3 increases the influence of tf or qtf respectively, whereas $k_1 = 0$ or $k_3 = 0$ vanishes with them. The constant b modifies the effect of document length: $b = 1$ is for cases where long documents are repetitive; $b = 0$ for cases where long documents are multi-topic, removing the length effect.

Our k-NN algorithm has four free parameters, k_1, k_3, b and k (neighbourhood size). We tune only the number of neighbours k so that it maximises the precision at the top rank document. It is performed via a 10-fold cross-validation process using the set of 976 English paper abstracts. Table 2 shows the k values found for the different classifiers. The Japanese abstracts are not used to determine different $k's$ for the J2E subtask given that before being submitted to the IR engine, abstracts are first translated into English. The parameter b is suggested by the Terrier experiments with the .GOV collection and set to 0.2681. We do not tune k_1 and k_3 parameters from Eq. (1). The default values of Terrier are used in this case: $k_1 = 1.2$ and $k_3 = 8.0$.

The classification workflow is depicted in Fig. 2; a query is provided to the IR engine, which ranks the first k documents d_j according to Eq. (1). The documents are substituted by their respective codes c_i and the codes are further re-ranked using to the methods described in the next subsection. A ranked list of n codes is then created. For the case of the J2E sub-subtask, the abstracts are first translated using Google Language Tools[4] before being used as input to the IR engine.

3.3 Re-ranking methods

In our attempt to improve the precision of the top n ranked codes, we have assessed several re-ranking algorithms analogously to the work of Xiao et al. [32] in the previous NTCIR challenge. They can be divided in simple and combined methods. And concerning the collections, they can be classified into simple and multi-collection. They are implemented as follows.

3.3.1 Simple methods

- Code frequency (codefreq) - In this ranking mode, the codes c_i are ranked according to their occurrence in the k patent neighbours d_j. Eq. (4) defines how the code's score are computed:

$$codefreq_{c_i} = \sum_{j=1}^{k} f(c_i, d_j), \tag{4}$$

[4]http://translate.google.com/

where f is defined by

$$f(c_i, d_j) = \begin{cases} 1, & \text{if } c_i \in d_j; \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

- Document similarity (docsim) - Here, the similarity of the k documents d_j are assigned to their respective codes c_i, as described in (6):

$$docsim_{c_i} = \sum_{j=1}^{k} f(c_i, d_j), \tag{6}$$

where f is defined by

$$f(c_i, d_j) = \begin{cases} sim(d_j), & \text{if } c_i \in d_j; \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

and the similarity $sim(d_j)$ of the document to the query is given by Eq. (1). In this method, the similarity of a document to the query can be directly assigned to a code (*docsim*) but also the average similarity (*docsim$_{avg}$*), which is given by dividing the document similarity by the number of codes it contains.

- Citation (citation) - It has been shown in previous experiments that citation is an important source of information for patent classification [1, 2]. How the codes c_i are scored according to this method is presented in Eq. (8):

$$citation_{c_i} = \sum_{j=1}^{k} f\left(c_i, g(d_{c,j}, d_j)\right), \tag{8}$$

where f is defined by:

$$f\left(c_i, g(d_{c,j}, d_j)\right) = \begin{cases} sim(d_j), & \text{if } c_i \in d_{c,j} \\ & \text{and } d_{c,j} \in d_j; \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

and $d_{c,j}$ is a document cited by d_j. Analogously to the *docsim* method, the citation can have simple (*citation*) and average (*citation$_{avg}$*) methods, which is given by dividing the similarity of the neighbour document by the number of documents it cites. Unfortunately, from our collections only the USPTO documents provide bibliography information.

3.3.2 Combined methods

- Normalised (norm) - Since the k-NN algorithm tends to give higher ranks to the most common codes in the collection, we try to reduce the impact of this effect by adding the code's frequency in the whole collection into the k-NN algorithm, as shown in Eq. (10):

$$norm_{c_i} = \alpha^{\frac{codefreq_{c_i} docsim_{c_i}}{\log(ffactor_{c_i} k)}}, \tag{10}$$

where α is a tuning parameter , k is the number of nearest neighbours and $ffactor_{c_i}$ is a function of the code's occurrence in the collection as defined in the following equation:

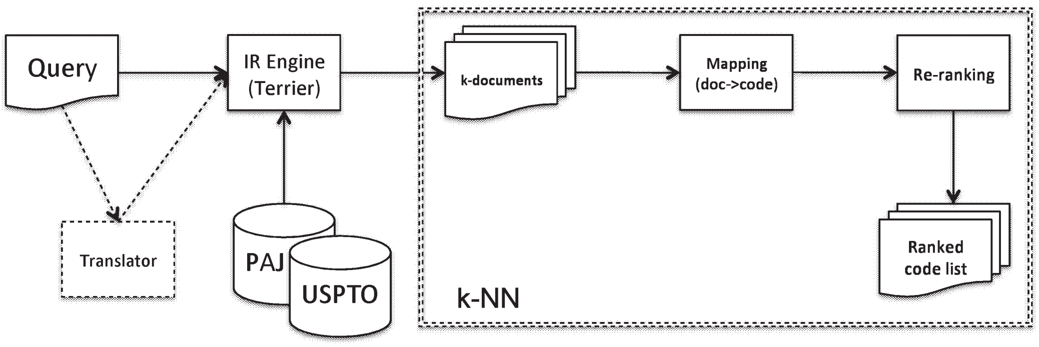

Figure 2: Classification workflow with the system 3 main components: query translator, IR engine and the k-NN classifier.

$$ffactor_{c_i} = \sum_{j=1}^{n} f(c_i, d_j). \qquad (11)$$

where f has the same definition of Eq. (7). Eq. (11) differs from Eq. (6) only in the sense that it iterates through the whole collection ($d_j \in c$) rather than only through the top k ranked documents ($d_j \in c_k$).

We use $\alpha = 1 + 10^{-10}$ in the classifiers. The value is chosen empirically so that it cushions very high and very low values in the exponent.

- Combined (combined): Here, we do a linear combination of the methods proposed in Eq. (4), (6), (8) and (10). Since, they are linearly independent - or have independent errors - their combination tends to outperform an individual classifier [23]:

$$combined_{c_i} = docsim_{c_i} + \alpha codefreq_{c_i} + \beta norm_{c_i} + \gamma citation_{c_i}. \qquad (12)$$

Again, values of the coefficients α, β and γ were not trained but rather chosen empirically to be $\alpha = \beta = \gamma = 0.01$.

- Multi-collection (multicoll) - We also tested combining the results of the classifiers proposed in Eq. (4), (6), (8) and (10) using the different indexed collections:

$$multicoll_{c_i} = \alpha score_{pajc_i} + \beta score_{usptoc_i} + \gamma score_{uspto\_claim_{c_i}}, \qquad (13)$$

where $score_{collection}$ is one of the methods $codefreq$, $docsim$, $citation$ or $norm$.

3.4 Named Entity Recogniser design

We choose the Conditional Random Field (CRF) framework [14] to provide the Named Entity Recogniser (NER) system. Mallet[5] and the OpenNLP[6] packages are used in the system's design. The OpenNLP is used for segmenting, tokenisation and part of speech tagging. Three different models are created and trained using the Sequence Tagging package of Mallet for each one of the different tags: technology, attribute and value. The effect tag is annotated via regular expression, matching any sequence of <*VALUE*> ... </*ATTRIBUTE*> or <*ATTRIBUTE*> ... </*VALUE*> in the text. We also segment the abstract documents into *title* and *abstract* fragments and train different models for them. The motivation behind this is the presence of only technology phrases in the title segment while in the abstract, the four tags can be found. Additionally, they have different syntactic structure.

Table 3: Technical trend map models.

features \ model	token	token and ps	all
token	x	x	x
has_previous_token	x	x	x
has_next_token	x	x	x
part_of_speech	-	x	x
has_previous_ps	-	x	x
has_next_ps	-	x	x
paragraph_size	-	-	x
paragraph_position	-	-	x
sentence_features	-	-	x
sentence_length	-	-	x
sentence_parenthesis	-	-	x
sentence_punctuation _marks	-	-	x
sentence_position	-	-	x
is_capital	-	-	x
is_alphanumeric	-	-	x
is_in_counter_part	-	-	x

The first model, *token*, takes into account only three features: token, previous token and next token. The second, *token_and_ps* extends the first by adding the part of speech elements *part of speech*, *previous part of speech* and *next part of speech*. The last model, *all*, contains 16 features. Additionally to features found in the *token_and_ps* model, it contains the *sentence's paragraph size*, *sentence's paragraph position*, *number of sentence's features*, *sentence length*, *number of sentence's parenthesis*, *number of sentence's punctuation marks*, *token's sentence position*, *is token capital*, *is token alphanumeric* and *is token in counter part* (if an abstract's token is found in the title and vice-versa). Table 3 contains a resume of the features used to create the models.

The trained models are complemented by a set of rules which tries to improve the precision of the NE recogniser:

[5]http://mallet.cs.umass.edu/
[6]http://opennlp.sf.net/

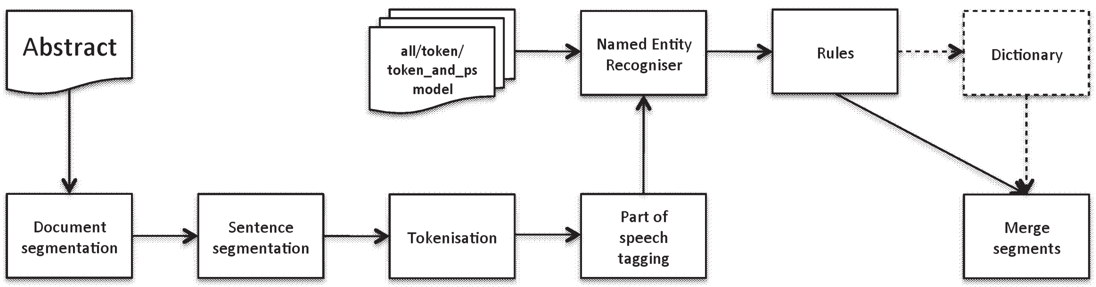

Figure 3: Technical trend map workflow.

- *isLinkWord*: if the phrase contains one of the link words "and", "or", "by", "for" or "with", the link word is removed. The phrase is split if the token is not at the begin or end of the sentence;
- *hasRelevantWord*: if the tagged phrase does not have any relevant word, i.e. all words are stop-words, the phrase is not tagged;
- *isOpenParenthesis*: if the first word of a tagged phrase is preceded by a parenthesis, the parenthesis is added to the phrase;
- *isCloseParenthesis*: if the last word of a tagged phrase is followed by a parenthesis, the parenthesis is added to the phrase;
- *isEndTag*: if the next token of a sequence to be tagged is a punctuation mark, the tag is closed and a new one may open after the punctuation mark;
- *isFalseTech*: for the technology tagger, if the tagged phrase is composed by only one word such as "recently", "can", "methods", "dynamic", "models" or "images", the phrase is discarded.

Finally, the technology/effect dictionary provided in the training set is used during the tagging process. The dictionary contains 1778 *technology*, 493 *attribute* and 287 *value* phrases. Ideally, we need to assess the terms individually and remove the noise from the dictionary. However, in this case the dictionary is composed of every tagged element found in the training sets. For the *value* phrases, we substitute numbers for regular expression and remove pure numbers from the dictionary.

Fig. 3 shows the workflow of the technology/effect annotation system. First, an "abstract" document (which in fact contains title and abstract sections) is split into title and abstract. Then, a sentence segmentation process is applied to them. The sentences are broken down into tokens and every token is associated to a part of speech tag. This information is submitted to the NE recogniser which will tag every token according to one of the three trained models. The tagged tokens are reconstructed using the previously described handcrafted rules. Finally, the title and abstract segments are merged to create the original paper/patent abstract document.

In the next section we report on the results we obtain in the *RPC* and *TTMC* subtasks.

4. RESULTS

In the challenge's formal run, the *RPC* system is tested with 633 English paper abstracts in the English sub-subtask and with 639 Japanese papers abstracts in the J2E. In the *TTMC*, 200 paper and 200 patent abstracts are used to assess the system's performance. The results of the *RPC* subtask are evaluated using the Mean Average Precision. On the other hand, for the *TTMC*, the results are reported using the F-score measure. The evaluation tool for this subtask considers correct only exact matches in a tagged passage, i.e. it is a binary assessment mode. The workshop organisers provide both evaluation tools.

In our experiments in the *RPC* subtask, we work with the classification of English and Japanese paper abstracts using the PAJ and USPTO collections. For each sub-subtask, we submitted three official runs for each one of the three levels of the IPC which are being tested in the challenge: *SC_BiTeM_sim*, *SC_BiTeM_weak* and *SC_BiTeM_combined* for subclass classification; *MG_BiTeM_sim*, *MG_BiTeM_weak* and *MG_BiTeM_combined* for main group; and *SG_BiTeM_sim*, *SG_BiTeM_weak* and *SG_BiTeM_combined* for subgroup. The *sim*, *weak* and *combined* results are obtained using Eq. (6), (10) and (12) respectively. For *weak* and *combined* methods only the PAJ collection is used while for *sim*, the codes are re-ranked using multi patent collections with $\alpha = 1$, $\beta = 0.1$ and $\gamma = 0.01$ in subclass and subgroup classifiers and $\alpha = 1$, $\beta = 0.1$ and $\gamma = 0$ in the main group.[7]

For the English sub-subtask, our best submitted runs obtain 0.6833, 0.4971 and 0.2991 of performance (MAP) when classifying paper abstracts according to subclass, main group and subgroup respectively. While for the J2E sub-subtask, these values are 0.7051, 0.5001 and 0.3028 for the corresponding IPC levels. The best official scores are all obtained using the *docsim* (*sim*) approach through a combination of collections.

The results of the methods proposed in Subsection 3.3 are resumed in Table 4 for the English sub-subtask. The column *"multi-coll parameters"* contains the values of α, β and γ of Eq. (13). For example, $1; 0; 0$ in the first row means $\alpha = 1$, $\beta = 0$ and $\gamma = 0$, which is the case where only the PAJ collection is used.

From the table, we can notice that for the methods *codefreq*, *docsim*, *norm* and *combined* there is small statistically significant difference between them, with a modestly enhanced performance for the *combined* method. However when it goes from one collection to another, there is a big difference in the results. As we can see, the USPTO collection tends to degrade the results, as previously reported

[7]The main group official run is not displayed in Table 4.

Table 4: Results of the experiments in the English subtask. †Official results submitted. ‡Since the PAJ collection does not contain citation information, these results come from the USPTO and USPTO_CLAIM indexed collections.

Classifier	*codefreq*	*docsim*	*norm*	*citation*	$citation_{avg}$	*combined*	*multi-coll* parameters
subclass	0.6651	0.6653	0.6612†	-	-	0.6660†	1;0;0
main group	0.4785	0.4799	0.4689†	-	-	0.4799†	1;0;0
subgroup	0.2853	0.2854	0.2819†	-	-	0.2857†	1;0;0
subclass	0.6001	0.6015	0.5995	0.5943	0.5871	0.6045	0;1;0
main group	0.3919	0.3905	0.3890	0.3563	0.3442	0.3951	0;1;0
subgroup	0.1656	0.1694	0.1683	0.1613	0.1534	0.1729	0;1;0
subclass	0.5914	0.5928	0.5968	0.5938	0.5825	0.5965	0;0;1
main group	0.3871	0.3878	0.3877	0.3470	0.3449	0.3917	0;0;1
subgroup	0.1577	0.1625	0.1623	0.1606	0.1523	0.1660	0;0;1
subclass	0.6932	0.6833†	0.6845	0.5951‡	-	-	1;0.1;0.01
main group	0.5008	0.4978	0.4777	0.3572‡	-	-	1;0.1;0.01
subgroup	0.3107	0.2991†	0.2870	0.1623‡	-	-	1;0.1;0.01

[32]. Moreover, we can notice that the combination of collections (*multi-coll* method) for one of the methods *codefreq*, *docsim* or *norm* always outperforms a single collection by at least 1.8% up to 4.2% in relative values. Another observation is the performance reached by the *citation* method. For the cases where it is possible to compare (USPTO and USPTO_CLAIM indexes), this method is very competitive with the others proposed (apart from the *maingroup* classifier). Finally, our best run (*multi-coll codefreq*) is not part of the official submission. The comparison of the J2E results shows a similar picture to the one depicted in Table 4 and corroborates with the aforementioned remarks.

For the *TTMC* subtask, we submitted four official runs for each one of the English sub-subtasks. The first three runs, officially labelled *paper_patent_BiTeM_1*, *paper_patent_BiTeM_2* and *paper_patent_BiTeM_3*, are based on the three models previously described in Subsection 3.4, *all*, *token* and *token_and_ps* respectively. All these runs use the built in dictionary in addition to the NER engine. The run *paper_patent_BiTeM_4* is also based on the *all* model but unlike *paper_patent_BiTeM_1* it does not use the complementary dictionary.

In our best runs, we obtain an *F-score* of 0.138 and 0.141 in the patent and paper sub-subtasks respectively both using the *token_and_ps* model aided by the dictionary. Our paper results prove to be very competitive when compared to the other groups especially when recognising *technology* phrases in title and *value* in abstracts. Our results of the patent *TTMC*, which are statistically the same as the ones achieved in the paper sub-subtask, are relatively lower than what is reported by other groups. The average[8] F-score results from all participants in the *TTMC* subtask are shown in Table 5.

Comparing *paper_patent_BiTeM_4* and *paper_patent_BiTeM_1* in Table 5, we see that the use of the dictionary does improve the system's performance by about 85%. We believe that the use of an ad-hoc dictionary, created from the training set, can improve results even further once noisy terms are removed from it. In the official run, we have not tested how much the rules created improve the system performance. However, during the training phase we observed a small performance gain.

[8]The average F-score measure does not take into account the *effect* tag.

In the previous results, the models are created using only data of the specific subtask, i.e. 300 abstracts per model. We have created another three models similar to them (*token*, *token_and_ps* and *all*) but this time using the whole training set. When the system is evaluated using these models, the best results of the official run are slightly improved, reaching an F-score of 0.148 for the patent sub-subtask and 0.146 for the paper task. Both of them use the *token_and_ps* model.

Fig. 4 shows the results of our best official runs in the paper and patent sub-subtasks. As mentioned before, tagging *technology* in title segments and *value* in abstracts are more effective than tagging *technology* and *attribute* in abstracts. We believe that the better performance on the *technology* in title is due to the simpler syntactic structure of this section when compared to the abstract. On the other hand, the improvement of the *value* tagging precision comes from the use of the dictionary. When comparing *paper_patent_BiTeM_4* with *paper_patent_BiTeM_1*, the performance of all other elements but *value* are relatively similar. The latter is up to 200% better than what is obtained when using the dictionary (F-score of 0.063 and 0.212 respectively for patent abstracts).

5. CONCLUSIONS

From the results presented in the previous section, we notice that the quality of the training set is fundamental in the classification task. In the USPTO collection, besides the wider coverage of the IPC taxonomy than the PAJ collection, as we can see from Table 1, more than half of the subgroup codes are found in fewer than 10 documents. This can explain the difference in performance of the classifiers that use the PAJ collection and the ones that use the USPTO. Moreover, the training set provided in the TTMC subtask contains tags like the following:

```
...<TECHNOLOGY>bead supports (S)</TECHNOLOGY>...
...<TECHNOLOGY>input terminals (4a, 4b)
</TECHNOLOGY>....
```

where non-technology terms *(S)* and *(4a, 4b)* are tagged as *technology*, while they are probably references to figures in the paper or patent document. We have not assessed how

Table 5: Average F-score results from all participants in the TTMC subtask.

System	Paper Avg F-score	Patent Avg F-score
paper_patent_NUSME-3	0.164	0.332
paper_patent_BiTeM_3	0.141	0.138
paper_patent_BiTeM_2	0.139	0.135
paper_patent_NUSME-2	0.132	0.296
paper_patent_BiTeM_1	0.130	0.135
paper_patent_ISTIC-1	0.110	0.295
paper_patent_ISTIC-3	0.102	0.165
paper_patent_ISTIC-2	0.076	0.295
paper_patent_BiTeM_4	0.070	0.107
paper_patent_NUSME-1	0.051	0.204
paper_patent_KAIST-IRNLP	0.031	0.018
patent_ISTIC-1-1	-	0.309
patent_ISTIC-2-1	-	0.292

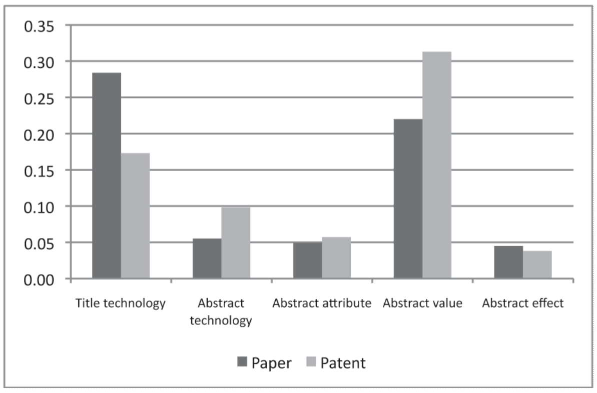

Figure 4: Individual results for the best runs of the TTMC subtask.

much it can interfere in the NER process.

In the *RPC* subtask we use a smaller sample of the patent collections to perform the IR task. Instead of the approximately 4.5 million documents provided in the PAJ and USPTO collections, the collections indexed have about 3 million documents in total. It is possible that this may have degraded our results in the classification subtask.

Several parameters of our system are not optimally tuned in the RPC (*norm*, *combined* and *multi-coll* constants) and TTMC (dictionary) subtasks. This could affect the performance of the classifiers and the NER. For instance, we expect the *combined* method in the classification subtask to significantly outperform the other methods however is not the case.

Finally, the lack of time stamps in the abstracts in the *TTMC* generation does not allow actual trends to be generated. Additionally, since the task is not focused in one specific *technology* field and the low recall of our system, it is difficult to correlate *effects* to a given technology using the test collection. From our results, only two technologies appear in more than one abstract and only one of them has *effect* tagged in two different articles. So, even if dates were provided we would not be able track any trend.

In this paper we report our experiments in the Patent Mining task of the NTCIR-8 challenge. Our group participates in the Research Papers Classification (English and J2E) and Technical Trend Map Creation (paper and patent) subtasks. In the *RPC* subtask, the *subclass*, *main group* and *subgroup* classifiers achieve a MAP of 0.6833, 0.4971 and 0.2991 respectively in the English sub-subtask. These values go to 0.7051, 0.5001 and 0.3028 for the J2E sub-subtask. For the *TTMC* subtask, the system achieves an F-score of 0.141 when tagging technology/effect elements in paper abstracts and 0.138 in patent abstracts. The use of the multi-patent collections improved significantly the performance of the classification system. The same is observed for the use of the dictionary to detect technology/effect phrases in the abstracts.

We plan to use all the documents from the USPTO and PAJ collections to see if we can further improve our classification results. Moreover, we want to exercise the classification system using bigrams. For the NER, we want to improve the existing rules and create new ones. Furthermore, the noisy terms in the dictionary needs to be removed. Finally, we hope such experiments can be beneficial for *ad hoc* retrieval in patent collections [6], in particular for chemical information searching [5].

6. ACKNOWLEDGMENTS

This research has been supported by the EU-IST-FP7 DebugIT project # 712139.

7. REFERENCES

[1] S. Chakrabarti, B. Dom, R. Agrawal, and P. Raghavan. Using taxonomy, discriminants, and signatures for navigating in text databases. In *Proceedings of 23rd VLDB conference*, 1997.

[2] S. Chakrabarti, B. Dom, R. Agrawal, and P. Raghavan. Scalable feature selection, classification and signature generation for organizing large text databases into hierarchical topic taxonomies. *The VLDB Journal*, 7:163–178, 1998.

[3] C. Fall, A. Töresvári, K. Benzineb, and G. Karetka. Automated categorization in the international patent classification. *ACM SIGIR Forum*, 37(1):10–25, 2003.

[4] F. C. Gey and R. R. Larson. Patent mining: A baseline approach. In *Proceedings of NTCIR-7 Workshop Meeting*, 2008.

[5] J. Gobeill, D. Teodoro, E. Patsche, and P. Ruch. Report on the trec 2009 experiments: Chemical ir track. In *The Eighteenth Text REtrieval Conference (TREC-18)*, 2009.

[6] J. Gobeill, D. Teodoro, E. Patsche, and P. Ruch. Simple pre and post processing strategies for patent searching in the clef intellectual property track 2009. In *CLEF Processing*, 2010.

[7] A. Hu and A. Jaffe. Patent citations and international knowledge SSow: the case of korea and taiwan. *Int. J. Ind. Organ.*, 21:849–880, 2003.

[8] A. Jaffe, M. Trajtenberg, and M. Fogarty. Knowledge spillovers and patent citations: evidence from a survey of inventors. *Am. Econ. Rev*, 90(2):215–218, 2000.

[9] M. Karki. Patent citation analysis: a policy analysis tool. *World Pat. Inf.*, 19(4):269272, 1997.

[10] Y. G. Kim, J. H. Suh, and S. C. Park. Visualization of patent analysis for emerging technology. *Expert Systems with Applications*, 34:1804–1812, 2008.

[11] T. Kohonen, S. Kaski, K. Lagus, J. Salojärvi, J. Honkela, V. Paatero, and A. Saarela. Self organization of a massive document collection. *IEEE Transactions on Neural Networks*, 11(3):574–585, 2000.

[12] M. Krauthammera and G. Nenadic. Term identification in the biomedical literature. *Journal of Biomedical Informatics*, 37:512–526, 2004.

[13] M. Krier and F. Zaccà. Automatic categorization applications at the european patent office. *World Patent Information*, 24:187–196, 2002.

[14] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: probabilistic models for segmenting and labeling sequence data. In *In Proceedings of the Eighteenth International Conference on Machine Learning (ICML-2001)*, 2001.

[15] L. S. Larkey. Some issues in the automatic classification of u.s. patents. In *Working Notes for the Workshop on Learning for Text Categorization, 15th Nat. Conf. on Artif. Intell. (AAAi-98)*, Madison, Wisconsin, 1998.

[16] B. Lent, R. Agrawal, and R. Srikant. Discovering trends in text databases. In *KDD-97 Proceedings*, 1997.

[17] Y. Mizuta and N. Collier. Zone identification in biology articles as a basis for information extraction. In *Proceedings of the joint NLPBA/BioNLP Workshop on Natural Language for Biomedical Applications*, pages 119–125, 2004.

[18] H. Nanba, A. Fujii, M. Iwayama, and T. Hashimoto. Overview of the patent mining task at the ntcir-7 workshop. In *Proceedings of NTCIR-7 Workshop Meeting*, 2008.

[19] H. Nanba, A. Fujii, M. Iwayama, and T. Hashimoto. Overview of the patent mining task at the ntcir-8 workshop. In *Proceedings of NTCIR-8 Workshop Meeting*, 2010.

[20] H. J. No and Y. Park. Trajectory patterns of technology fusion: Trend analysis and taxonomical grouping in nanobiotechnology. *Technological Forecasting & Social Change*, 77:63–75, 2010.

[21] S. Robertson, S. Walker, M. M. Beaulieu, M. Gatford, and A. Payne. Okapi at trec-4. In *The Fourth Text REtrieval Conference (TREC-4)*, 1995.

[22] P. Ruch, I. Tbahriti, J. Gobeill, and A. R. Aronson. Argumentative feedback: A linguistically-motivated term expansion for information retrieval. In *ACL*, pages 835–40, 2006.

[23] F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34:1–47, 2002.

[24] T. Shimano and T. Yukawa. An automated research paper classification method for the ipc system with the concept base. In *Proceedings of NTCIR-7 Workshop Meeting*, 2008.

[25] L. Tanabe and W. J. Wilbur. Tagging gene and protein names in biomedical text. *Bioinformatics*, 18(8):1124–1132, 2002.

[26] D. Tikk and G. Biró. Experimental results with hitec on various document corpora. In *In Tutorial Proceedings of the Information Technology Research Center (ITRC) Forum*, 2002.

[27] D. Tikk, G. Biró, and A. Törcsvári. A hierarchical online classifier for patent categorization. In *Emerging Technologies of Text Mining: Techniques and Applications*. Idea Group Inc., 2007.

[28] A. J. C. Trappey, F.-C. Hsu, C. V. Trappey, and C.-I. Lin. Development of a patent document classification and search platform using a back-propagation network. *Expert Systems with Applications*, 31:755–765, 2006.

[29] Y.-H. Tseng, C.-J. Lin, and Y.-I. Lin. Text mining techniques for patent analysis. *Information Processing and Management*, 43:1216–1247, 2007.

[30] I. von Wartburg, T. Teichert, and K. Rost. Inventive progress measured by multi-stage patent citation analysis. *Res. Policy*, 34:1591–1607, 2005.

[31] W. Wang, S. Li, and C. Wang. Icl at ntcir-7: An improved knn algorithm for text categorization. In *Proceedings of NTCIR-7 Workshop Meeting*, 2008.

[32] T. Xiao, F. Cao, T. Li, G. Song, K. Zhou, J. Zhu, and H. Wang. Knn and re-ranking models for english patent mining at ntcir-7. In *Proceedings of NTCIR-7 Workshop Meeting*, 2008.

[33] Y. Yang. An evaluation of statistical approaches to text categorization. *Information Retrieval*, 1:69–90, 1999.

[34] Y. Yang and X. Liu. A re-examination of text categorization methods. In *Proceedings of SIGIR'99 conference*, 1999.

[35] U. Zernik. Method for tagging collocations in text, 1995.

[36] D. Zhua and A. L. Porter. Automated extraction and visualization of information for technological intelligence and forecasting. *Technological Forecasting & Social Change*, 69:495 – 506, 2002.