Scientometrics (2019) 121:1239–1268
https://doi.org/10.1007/s11192-019-03246-1

Analysis of the effect of data properties in automated patent classification

Juan Carlos Gomez[1]

Received: 4 April 2018 / Published online: 3 October 2019

Abstract

Patent classification is a task performed in patent offices around the world by experts, where they assign category codes to a patent application based on its technical content. Nowadays, the number of applications is constantly growing and there is an economical interest on developing accurate and fast models to automate the classification task. In this paper, we present a methodology to systematically analyze the effect of three patent data properties and two classification details on the patent classification task: patent section to use for training/testing, document representation, patent codes to use for training, use of the hierarchy of categories, and the base classifier. For the analysis we create a diversity of models by combining different options for the properties. We evaluate the models in detail using standard patent datasets in two languages, English and German, considering three performance metrics, using statistical tests to validate the results and comparing them with other models in the literature. Our research findings indicate that it is important to follow a methodology to properly choose the options for the data properties to build a model according to our goal, considering classification accuracy and computational efficiency. Some combinations of options build models with good results but with high computational cost, whilst other build model that produce slightly worst results but at a fraction of the training time.

Keywords Patent classification · Hierarchical classification · Multilabel classification · Document representation · Supervised learning · IPC

Introduction

Patent classification is one of the first tasks performed by experts of patent offices when analyzing a *patent application* to register a new invention. Classification consists in assigning a set of category codes to the document, based on their content, ensuring in this way that patents with similar topics or technological areas are grouped under the same codes. Accurate classification of patent applications is vital for the inter-operability between different patent offices, and for conducting several tasks such as reliable patent search,

✉ Juan Carlos Gomez
jc.gomez@ugto.mx

[1] Departamento de Ingeniería Electrónica, DICIS, Universidad de Guanajuato, Salamanca, Mexico

management and retrieval (Zhang et al. 2015; Abbas et al. 2014); extract relevant content (Härtinger and Clarke 2015; Rodriguez-Esteban and Bundschus 2016; Wang et al. 2014); and investigate technology characteristics (Arts et al. 2018; Noh et al. 2015).

Nonetheless, the amount of patent application fillings is constantly growing and they are overwhelming the experts who analyze the documents. For example, the number of patent applications received by the United States Patent and Trademark Office (USPTO) in 2015 amounted to 629,647[1], whilst the European Patent Office (EPO) received approximately 296,227 patent applications in 2016[2].

Additional to the great amount of data, patent data have particularities that play a role when doing the classification. First, the categories are organized as a hierarchy, and this hierarchical structure is large and complex (containing thousands of categories in a tree-like structure). Second, patents are in turn complex and lengthy documents, composed by several pages and divided by sections. Third, patents are usually labelled with several categories at the same time, meaning they comprise different technological areas. If we additionally consider that experts are costly and vary in capabilities when performing the classification, it is clear that there is a necessity for reliable and efficient automatic methods to help in the patent classification task.

The automated patent classification task has been tackled along the last decade as a text classification problem using several methods and approaches (Benzineb and Guyot 2011; Gomez and Moens 2014). Nevertheless, despite the research done it is still an open problem with several unsolved issues regarding the general low accuracy obtained (Benzineb and Guyot 2011; D'hondt et al. 2017; Fall and Benzineb 2002; Gomez and Moens 2014). In this paper, we aim to contribute towards gaining more understanding of the problem by proposing a methodology to systematically study the effect on classification of three patent data properties in combination with two general text classification details. The properties we study are the use of: the different sections to extract content from patents, the different codes assigned to each patent, and the hierarchy of categories. The two details in classification we consider are: the way to represent documents and the base classifier.

Following our methodology, we train and test optimized classification models using different combinations of options for the mentioned properties. We then compare the results of applying the models over two standard patent datasets in English and German. We consider classification accuracy and computational efficiency, and conduct statistical tests to assert the validity of the comparisons. Additionally, we show comparisons with other works in the literature, considering the methodologies, the models and the results. In addition, we also conduct a statistical characterization of the datasets.

Our contributions for the problem of patent classification are five. (1) A methodology that take into account several relevant patent data properties and classification details for the study of the problem. (2) A thorough systematic analysis of the effect in classification results of different models built using the methodology as combinations of patent data properties. (3) The consideration of two important aspects of the problem, the optimization of hyper-parameters in the base classifiers and the language independence of the models. (4) Introduction of a hierarchical model that train local classifiers and compute the final classification as a weighted linear combination of the decisions along the hierarchy. (5) Use of our findings as a guideline for the patent classification task, such that other researchers

[1] http://www.uspto.gov/.

[2] https://www.epo.org/about-us/annual-reports-statistics/statistics.html.

Table 1 Example of a sequence of codes in the IPC

IPC	Code	Title
Section	H	Electricity
Class	H01	Basic electric elements
Subclass	H01F	Magnets
Main group	H01F 1/00	Magnets or magnetic bodies
Subgroup	H01F 1/01	Of inorganic material
Subsubgroup	H01F 1/03..	Characterised by their coercivity

will consider them when making decisions about what options would be more suitable for implementing or for testing their hypotheses, considering classification accuracy and computational efficiency. Some combinations of options help to build models that are slow to train but produce good classification results, while other produce slightly worst results at a fraction of the training time.

In the following sections we first describe the patent data properties ("Patent data properties" section). Later, we review the relevant related works in the literature of the problem ("Relevant related works" section). We then describe in detail our experimental methodology ("Experimental methodology" section) with the several options for each property and classification detail and the experimental setup. Afterwards, we present the results ("Experimental analysis" section). We conclude our work in "Conclusions" section with an overall discussion and possible future research directions.

Patent data properties

The first property of patent data is that the categories to classify patents are organized as a tree-like hierarchical structure. There exist several structures used by different patent offices, but the most widely used and globally agreed is the International Patent Classification[3] (IPC), used by more that 100 countries with around 95% of all existing patents classified according to it. The World Intellectual Property Organization (WIPO)[4] manages and updates annually the IPC, being IPC2018.01 the current version.

Every category in the IPC has a code and a title name. The IPC divides all technological fields in eight sections, designated by capital letters from A to H. Each section is subdivided in classes, labeled by the section code followed by two digits (e.g. H01). Each class is divided in subclasses, labeled by the class code followed by a capital letter (e.g. H01F). Each subclass is broken down in main groups, labeled by the subclass code followed by a one to three digits, an oblique stroke and the number 00 (e.g. H01F 1/00). Subgroups form subdivisions under the main groups. Each subgroup code includes the main group code, but replaces the last two digits by other than 00 (e.g. H01F 1/01). Subgroups are ordered in the structure as if their numbers were decimals of the number before the oblique stroke. For example, 1/036 is to be found after 1/03 and before 1/04. After subgroup level, the hierarchy is organized using dots preceding the title of the category (e.g. H01F 1/03.),

[3] http://www.wipo.int/classifications/ipc/en/.

[4] http://www.wipo.int/portal/en/index.html.

Table 2 Number of categories in each level of the IPC (version IPC2018.01)

Level	Name	Categories
1	Section	8
2	Class	131
3	Subclass	642
4	Main group	7461
5 (and below)	Subgroup	66454

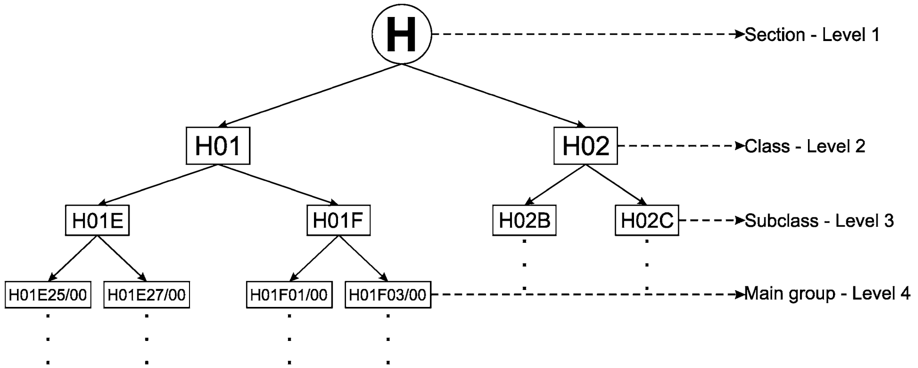

Fig. 1 Example of a portion of the IPC hierarchy starting at level 1, section B. The root node is level 0 (not shown)

where each dot represents a level down. An example of a sequence of category codes along the different levels of the IPC is shown in Table 1. The total number of categories per level in the IPC is shown in Table 2.

From a mathematical point of view, the IPC hierarchy is a directed rooted tree graph, with every code indicating a node. The nodes are connected by directed edges indicating PARENT–OF relationships, where each node has only one parent, meaning two nodes are connected by exactly one path. Figure 1 shows a portion of the IPC representing the tree graph. The root node is the level 0 and not shown.

The second property of patent data is that patents are complex documents (Zhang et al. 2015) and present differences with respect to other documents that are classified automatically, such as news, emails or web pages. Patents are long documents of several pages, their content is governed by legal agreements and is therefore semi-structured (divided by sections and with well-defined paragraphs), and are written using a formal language, with many technical words and sometimes fuzzy sentences (in order to avoid infringement of other patents or to extend the scope of the invention). The structure of a patent is important because it provides different input information to a classification model. The content of a patent is generally organized in the following sections (Fall and Benzineb 2002; Benzineb and Guyot 2011; Lupu and Hanbury 2013; Gomez and Moens 2014):

- Title: indicates a descriptive *name* of the patent.
- Bibliographical data: contains the number of the patent, the names of the inventor and the applicant, and sometimes the citations to other patents and documents.

- Abstract: includes a brief description of the invention presented in the patent.
- Description: contains a detailed description of the invention, including prior work, related technologies and examples.
- Claims: explains the legal scope of the invention and for which application fields the patent is sought.

Most of the sections contain pure text, but in a patent, it is also frequent to find images, graphics and links. In this work, we focus exclusively on the textual content, since it is the largest component in patents and several other elements in the content are often described or explained using text.

The third property of patent data is that patents can have more than one category code assigned to it, meaning it encompass several technological areas. The first code assigned by the experts corresponds to the most relevant category (main code). Secondary codes are related with other categories that are relevant for the patent, but without any specific order of relevance. From a machine learning perspective, the task is considered a multi-label problem (Tsoumakas et al. 2010).

Relevant related works

There are several works on the automated classification of patents, starting with some surveys about the task (Fall and Benzineb 2002; Krier and Zaccà 2002; Benzineb and Guyot 2011; Gomez and Moens 2014) where some issues are highlighted, such as model accuracy, scalability, use of the hierarchy of categories, patent sections to use, and document representations.

Fall et al. introduced the WIPO-alpha and WIPO-de datasets in Fall et al. (2003, 2004) respectively, where they performed a comparison of several base classifiers: NB, KNN, SVM, SNoW (sparse network of winnows) and LLSF (linear least squares fit), using different patent sections independently. They also introduced a set of performance metrics to evaluate the task. In Seneviratne et al. (2015) and Tikk et al. (2005) the authors presented several hierarchical models, using several base classifiers (such as SVM, KNN and HITEC), patent sections and patent codes, which they evaluated using the full WIPO datasets with the same performance as defined by Fall et al. We compare one of our models with the results in these works.

There are other works that have used the WIPO datasets (specially the WIPO-alpha) for experimentation, many of them focusing on kernel classifier methods that take the hierarchical structure into account (Bi and Kwok 2014; Cai and Hofmann 2004; Chen and Chang 2012; Rousu et al. 2006; Tsochantaridis et al. 2005; Zhang 2014). However, most of these works focus on assigning a single code to a patent, only conduct experiments on a subset of the dataset (normally on the Section D of the IPC), which make it unclear if they are scalable to the full dataset, or focus on a specific task, such as preferential classification.

Some works have used models based on neural networks such as back-propagation (Trappey et al. 2006) and Winnow (D'hondt et al. 2017; Koster et al. 2003) using different features such as phrases and deep learning word representations. The authors found that word features produce better results that other features in most of the cases.

Along the years, the NTCIR workshops have organized several patent classification tasks (Iwayama et al. 2005, 2007; Kim and Choi 2007; Nanba et al. 2008, 2010) to classify Japanese patents by the *F*-terms, or research publications in English in the IPC, based on

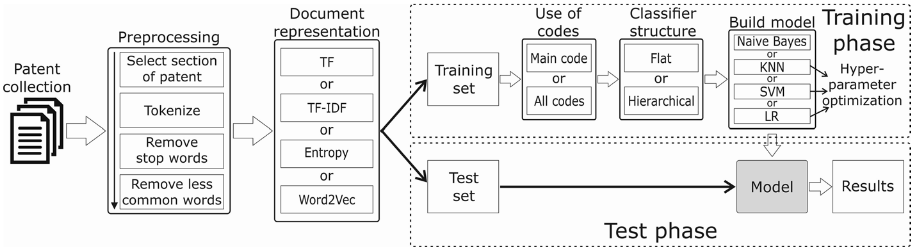

Fig. 2 General description of the experimental methodology

training with patent data. In Li and Shawe-Taylor (2007) the authors presented a series of methods based on SVM for cross-lingual patent classification between English and Japanese using the NTCIR-3 dataset.

CLEF-IP tracks included a task on patent classification Piroi (2010, 2011). The tasks provided a collection of over 1 million patents as training data to classify a test set of around 3000 patents at the subclass level of the IPC. Some of the best models were based on the Winnow classifier (Guyot et al. 2010; Verberne et al. 2010; Verberne and D'hondt 2011), using the abstract section and words and triplets features. In Giachanou and Salampasis (2014) and Giachanou et al. (2015) the authors used the CLEF-IP 2011 dataset to evaluate a series of methods based on information retrieval for patent classification, but using a subset of the test set of only 300 patents.

Some other works have used the WIPO, NTCIR, CLEF-IP or other patent datasets, but they used them as general text/graph datasets to conduct several forms of classification. These works had different goals than the particularities of patent classification, such as testing the efficiency and/or scalability of their particular methods in general text classification (Gomez and Moens 2014), hierarchical classification (Wang et al. 2014), or node classification in graphs (Dallachiesa et al. 2014); testing methods for extreme machine learning (Wang et al. 2014) or dimensionality reduction (Shalaby et al. 2014); quantifying the existence of concept drift in data (D'hondt et al. 2014); or classifying the data in user-defined hierarchies (Zhu et al. 2015). Most of these works used only the title, abstract or claims section from patents, used general accuracy and macro and/or micro-F1 as performance metrics, did not mention what patent codes they used, and considered or not the hierarchy depending on the problem they were studying.

Experimental methodology

Figure 2 shows a graphical depiction of our experimental methodology, which consists of several sequential phases composed of several steps or options.

We start with a collection of patents and the first phase of the methodology is preprocessing. The step of preprocessing consists in to choose the section (or combination of sections) to extract information from patents: title (T), inventors (I), abstract (A), claims (C), short description (S) (first 30 lines of the description) and long (full) description (L). The second step consists in tokenizing the content to extract word features as sequence of letters, numbers and hyphens (to capture chemical compounds) and convert each word to lowercase. The third and fourth step consist in removing words that carry little information. By

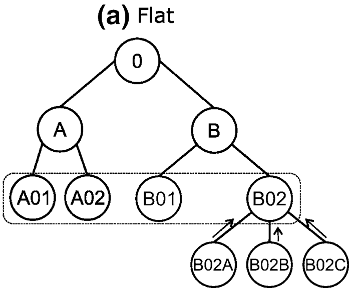

Fig. 3 Approaches of classification with multi-class classifiers in dashed squares. In the flat approach (**a**), the model predicts the local categories, in the hierarchical approach (**b**), the model predicts the children categories

default we remove stop words and words appearing in less than five training documents to form a vocabulary.

The second phase of the methodology is the document representation. This phase consists in indexing and transforming all the patent documents to a vector space representation using the vocabulary extracted in the previous phase. We considered four weighting options when vectorizing the patents: term-frequency (tf), term frequency–inverse document frequency (tf–idf), entropy and word2vec (w2v), and we normalized the document vectors using the L2 norm. The w2v features where computed using the Gensim module from Python[5] by training a skip-gram model (Mikolov et al. 2013) over the training part of each dataset (see next subsection), extracting word embeddings of 300 dimensions. The final document representation was computed by averaging all embeddings corresponding the its words.

We decided to use word features because in several works it have been pointed out that these features outperform other more complex representations in several classification/prediction tasks (Cinar et al. 2015; Basile et al. 2017), including patent classification (D'hondt et al. 2017).

In the third phase of the methodology, for training we can choose to include all the codes assigned to a patent or only the main code; specially considering that the multiple concurrent labels in a dataset could confuse a classification system (Fall et al. 2004; Tsoumakas et al. 2010).

The fourth phase of the methodology consists in deciding whether or not to use the hierarchical structure of the IPC. When the hierarchy is ignored (flat approach), for training we take all the categories at a defined level (section, class, subclass, main group), aggregates all the patents from the categories below, and build a single large multi-label classifier (Fig. 3a). During testing, the flat approach takes a test patent and returns all the categories from the defined level as a list ranked by probability.

For using the hierarchy (hierarchical approach), we introduce a model that trains a local multi-label classifier in each category that contains children categories, aggregating the patents from the children categories in the local classifier (Fig. 3b). During testing, our hierarchical approach takes a test patent and assigns codes from top to bottom in the hierarchy, starting at the section level. Each local classifier returns its local categories as a list

[5] Available at: https://radimrehurek.com/gensim/.

ranked by probability, and from each list the model takes the top three. It then goes down only in the selected categories. When going down, it adds the corresponding probabilities for the assigned categories weighted by an importance factor of the category level. When classifying to the main group level, we will have 81 ($3 \times 3 \times 3 \times 3$) selected codes. Each code would have a final probability P as:

$$P = aP_{\text{section}} + bP_{\text{class}} + cP_{\text{subclass}} + dP_{\text{maingroup}} \tag{1}$$

where P_{level} is the code probability for the given level and a, b, c and d are the level importance factors. These factors could modify the distribution of errors along the hierarchy. In our case, we established importance factors of 1/4 for all the levels. Thus, the final classification is a linear combination of the weighted decisions along the hierarchy.

In the fifth phase of the methodology, we choose what base classifier to use for building the local or global classifiers. We considered four options: Multinomial Naive Bayes (NB), a probabilistic classifier; K-Nearest Neighbors (KNN), a instance-based classifier; and linear Support Vector Machines (SVM) and Logistic Regression (LR), two discriminative classifiers. We used the implementation from WEKA (Hall et al. 2009) for NB, Liblinear (Fan et al. 2008) for SVM and LR, and for KNN a proprietary implementation that takes advantage of the sparseness of the vector representation. For models using SVM and LR we use L2 loss and regularization respectively, and for KNN we use 1-cosine similarity as the distance metric.

During the training phase, in our methodology we performed a fivefold cross validation over the training set for the models using KNN, SVM or LR, to look for the optimal number K of neighbors or the optimal soft margin parameter C respectively. We considered the values of $K = 1, 5, 10, 20$ and $C = 0.1, 1, 10, 100$, and we use the *top* metric (see below) as the optimality criterion. It is worth to mention that most of the works in the literature for patent classification do not perform a hyper-parameter optimization, but commonly takes the defaults values from the implementation.

For our experiments we trained a diversity of models using different combinations of the previous mentioned options: patent section to extract information, document representation, patent codes to use for training, use or not of the IPC hierarchy, and the base classifier. During the test phase, for each test patent, the flat models output all the possible categories, whilst the hierarchical models output a list of 81 codes, in both cases ranked by probability. For a matter of comparison, we choose the three top codes per test patent and evaluated each model using the performance metrics defined in Fall et al. (2003). The *top* metric compares the top predicted code with the main code of the test patent. The *three* metric compares the top three predicted codes with the main code of the test patent. The *all* metric compares the top predicted code with all the codes assigned to the test patent. To validate the results, we performed paired McNermar's tests between each pair of models, considering a significance level of $\alpha = 0.01$ and using the Holm–Bonferroni method to correct for the number of comparisons.

All the code for implementing the methodology was written in Java and Python and will be released upon acceptance. We conducted all the experiments using a desktop Linux PC with a 3.4 GHz Core i7 processor and 16 GB in RAM.

Table 3 Number of categories per level of the IPC, and number of codes per patent in the WIPO-alpha and WIPO-de patent datasets

Dataset	Number of categories per level of IPC				Codes per patent			
	Section	Class	Subclass	Main group	Min	Max	Avg	Std
WIPO-alpha	8	131	632	5907	1	25	1.88	1.43
WIPO-de	8	120	604	5627	1	12	2.05	1.27

Datasets description

We use for our experiments the standard patent collections introduced by Fall and Benzineb (2002), and Fall et al. (2004): WIPO-alpha, and WIPO-de. The WIPO-alpha[6] collection consists of 75,250 patent documents in English. The WIPO-de[7] consist of 110,826 patent documents in German provided by the German Patent Office and extracted from the DEPAROM[8] collection. Both datasets are stored in XML format and are already split in standard training and a test sets. Patents in both datasets include the sections: title, abstract (not present in the WIPO-de dataset), claims and the full description.

Table 3 shows the total number of IPC categories present in the WIPO-alpha and WIPO-de datasets, split per level of the IPC. We considered all the codes assigned to each patent for the statistics. The numbers in this table are consistent with the ones in Table 2, therefore the datasets are a good sub-sample of the whole IPC structure. Table 3 also shows the minimum, maximum, average and standard deviation of category codes assigned to each patent. The statistics are similar for both datasets, with around 95% of patents containing at most 5 codes.

Table 4 breaks down the number of codes for the training and test parts of each patent collection, and shows how the patents are distributed along the categories. The table shows that there are more categories in the training part than in the test part in both dataset. The columns of Max/Min and Ent contain information about the degree of skewness of the category distributions. The former shows the ratio between the major and minor categories; higher ratios result in more skewed category distributions. The latter shows the Shannon entropy values; higher values of entropy imply more uncertainty in the distribution. From the statistics, we observe that such distribution is largely skewed and with a high degree of uncertainty (especially at the class and subclass levels). Figure 4a and 4b present the distributions of patents per category at the main group level for both patent collections. We observe here the skewness of the distributions, for WIPO-alpha 56% of the categories contain at most 10 patents, while only 4% contain more than 100 patents. Similar distributions have been observed in other hierarchical structures (Gomez and Moens 2012).

For the experiments, we first extracted independently the content from sections: title, inventors, abstract, claims, short description, and full description, and also we extracted two combinations of all the sections, one with the short description (TIACS) and one with the long one (TIACL). TIACL combination corresponds to the full content of the patent. The TIACS and TIACL combinations for WIPO-de does not include the abstract

[6] http://www.wipo.int/classifications/ipc/en/ITsupport/Categorization/dataset/.

[7] See previous note.

[8] http://www.deparom.de.

Table 4 Statistics of documents over categories in the WIPO-alpha and WIPO-de datasets

Set	Level	WIPO-alpha								WIPO-de							
		Docs	Cats	Documents per category						Docs	Cats	Documents per category					
				Min	Max	Avg	Std	Max/min	Ent			Min	Max	Avg	Std	Max/Min	Ent
Training	Class	46,324	125	1	10,068	701.7	1215.0	10,068.0	6.7	84,820	120	5	11,556	1471.0	2181.4	2311.2	6.9
	Subclass		614	1	4513	142.9	316.7	4513.0	7.3		604	1	2695	292.2	420.6	2695.0	7.9
	Main group		5557	1	1721	15.8	35.2	1721.0	5.0		5542	1	852	31.9	63.4	852.0	5.9
Test	Class	28,926	127	1	6974	425.2	1044.1	6974.0	6.6	26,006	118	4	4036	425.4	775.1	1009.0	6.5
	Subclass		575	1	3226	93.9	270.0	3226.0	6.4		541	1	1038	92.8	176.5	1038.0	6.4
	Main group		4444	1	1189	12.2	48.1	1189.0	4.2		3948	1	373	12.7	28.3	373.0	4.5

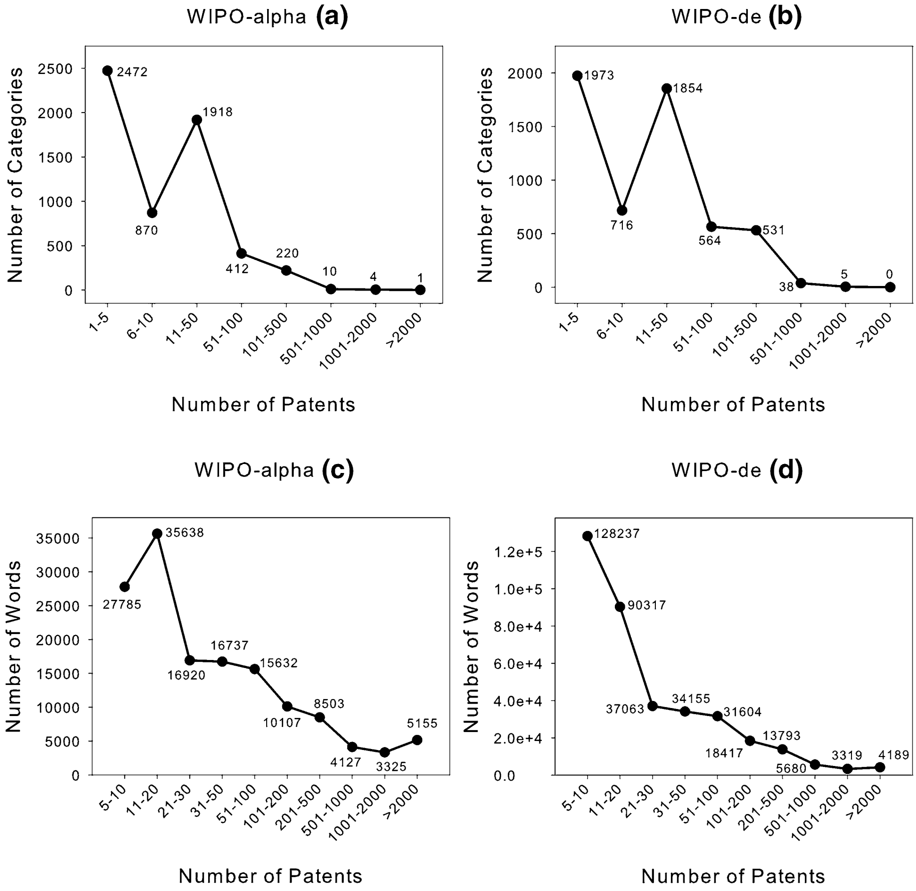

Fig. 4 Distributions of patents over categories, **a**, **b** words over patents, **c**, **d** WIPO-alpha and WIPO-de datasets

section (it is not present in the documents of the dataset). Table 5 shows the statistics on the number of (unique and total) words from the different extracted sections/combinations of both datasets. The full description is the largest individual section by far, followed by the claims. These two sections dominate the combinations TIACL and TIACS respectively. However, for these sections there is a high degree of repetition for some words, as indicated by the number of unique words.

Figure 4c and 4d shows the distribution of words in the TIACL combinations of both datasets. As in other large document collections, the word distribution in WIPO-alpha and WIPO-de follows the Zipf's law, with many words appearing in few documents and few words appearing in many documents. The number of words appearing in only one document (not included in the plots) in WIPO-alpha is 1,758,164 and in WIPO-de is 1,691,632. This could produce very large uninformative vocabularies, and thus filtering uncommon words is recommended. Table 6 shows the vocabulary size for each section/combination from both datasets (extracting from the training part).

Table 5 Statistics on number of words in each section and combination from the WIPO-alpha and WIPO-de datasets

Section	WIPO-alpha								WIPO-de							
	Total words				Unique words				Total words				Unique words			
	Min	Max	Avg	Std	Min	Max	Avg	Std	Min	Max	Avg	Std	Min	Max	Avg	Std
Title	1	32	4.8	2.4	1	21	4.7	2.2	1	51	5.4	3.7	1	33	5.2	3.3
Inventors	1	57	4.0	3.3	1	49	3.9	3.1	1	87	7.5	6.1	1	58	6.7	4.9
Abst	2	262	56.4	26.6	2	142	34.6	13.6	0	0	0	0	0	0	0	0
Claims	5	26,807	524.1	592.2	4	1494	98.1	52.5	1	13,074	392.4	320.9	1	1722	113.4	49.0
Desc (S)	12	574	204.0	42.8	10	302	115.3	25.8	8	396	144.3	38.0	5	222	91.3	20.8
Desc (L)	61	293,655	2993.8	4167.9	32	17,258	705.4	583.2	8	29,626	2154.1	1825.3	5	4871	570.7	308.3
TIACS	32	27,115	798.5	605.1	25	1872	200.0	62.8	1	13,446	523.6	341.7	1	1868	171.1	63.1
TIACL	33	297,825	3593.5	4443.7	26	17373	725.7	585.9	1	32,825	2424.9	2057.1	1	4900	564.6	331.5

Table 6 Number of words in the vocabulary of each section of the WIPO-alpha and WIPO-de datasets

Dataset	Vocabulary size							
	Title	Inventors	Abst.	Claims	Desc (S)	Desc (L)	TIACS	TIACL
WIPO-alpha	5469	5711	16,054	31,850	33,451	136,004	53,011	143,928
WIPO-de	9055	17,246	0	98,446	79,416	340,835	155,161	366,773

Experimental analysis

Use of the hierarchy

In our first experiment we compare the effect of using or not the IPC hierarchy and show the results in Table 7. For this, we use the WIPO-alpha dataset and the following setup: only the abstract section, a tf document representation, all the codes from patents and the four base classifiers. The first four rows in the table correspond to flat models and the last four to hierarchical models. Each row corresponds to one base classifier, with the optimal parameter (K or C) between parentheses. In the columns appear the results per level of the hierarchy (using the three performance metrics), and the training and test times (training time includes the cross-validation for finding the optimal parameter). We indicate in bold the best results for every metric and level. Letter A in supper index indicate values that are not significantly different than the best values in a column.

In this table, we observe that in both cases, with flat and hierarchical approaches, the best values are obtained by models that use LR with the same optimal parameter C. At the class and subclass levels, the hierarchical approach obtains better or similar values than the flat approach, but at the main group level, the flat approach obtains better results. This is consistent with findings for other hierarchical structures, such as web documents (Bennett and Nguyen 2009), since in the hierarchical approach, the errors are propagated and accumulated from top to bottom, and there is less information to discriminate because of the use of local classifiers. When analyzing the classification performance of the individual base classifiers, the models could be ranked in both approaches from best to worst as: LR, SVM, NB and KNN. Regarding training/test times for the base classifiers, NB presents almost no difference when using flat or hierarchical approaches, whilst for KNN the test time is similar on both approaches, but the flat approach is twice as fast as the hierarchical one for training. For KNN its training time corresponds to the cross-validation to find the optimal K. Most of the time in KNN is spent in calculating distances, and in the hierarchical approach, due to the set of local classifiers, it computes more distances than with the flat approach. Finally, SVM and LR have the biggest difference in training time between approaches, with hierarchical models being trained around 45 times faster than flat models, whilst their test times are comparable between approaches. This makes evident the advantage in efficiency of training local classifiers.

A first finding of this experiment is that the hierarchical approach in patent classification allows a faster training of models, but the flat approach creates models that have a better accuracy at the bottom level of the hierarchy. We could then decide if we want a better computational efficiency or a higher accuracy. A second finding is that our hierarchical approach, that builds local classifiers and compute the classification as a weighted linear combination of the decisions along the hierarchy, is well suited for the problem. A

Table 7 Performance comparison of the flat and the hierarchical approaches with the WIPO-alpha dataset

Structure	Classifier	Class			Subclass			Main group			Time (s)	
		Top	Three	All	Top	Three	All	Top	Three	All	Training	Test
Flat	NB	0.463	0.693	0.556	0.344	0.550	0.444	0.229	0.379	0.324	25	**53**
	KNN (20)	0.365	0.593	0.452	0.270	0.455	0.357	0.174	0.306	0.254	7183	3372
	SVM (0.1)	0.467	0.474	0.550	0.354	0.361	0.443	0.238	0.245	0.326	265,439	62
	LR (0.1)	0.505	0.734^{A}	0.602	0.391^{A}	**0.608**	**0.496**^{A}	**0.266**	**0.443**	**0.371**	268,544	72
Hierarchy	NB	0.452	0.660	0.524	0.316	0.467	0.393	0.167	0.256	0.231	**22**	56
	KNN (10)	0.345	0.584	0.425	0.256	0.412	0.338	0.158	0.240	0.235	14,858	3221
	SVM (0.1)	0.488	0.507	0.562	0.351	0.366	0.432	0.207	0.215	0.282	5968	59
	LR (0.1)	**0.521**	**0.738**^{A}	**0.615**	**0.394**^{A}	0.546	**0.496**^{A}	0.242	0.329	0.340	5489	54

Table 8 Performance comparison of different document representations and use of all codes or only main code with the WIPO-alpha dataset

Rep.	Classifier	Class			Subclass			Main group			Time (s)	
		Top	Three	All	Top	Three	All	Top	Three	All	Training	Test
All codes												
tf	NB	0.452	0.660	0.524	0.316	0.467	0.393	0.167	0.256	0.231	22	56
	KNN (10)	0.345	0.584	0.425	0.256	0.412	0.338	0.158	0.240	0.235	14,858	3221
	SVM (0.1)	0.488	0.507	0.562	0.351	0.366	0.432	0.207	0.215	0.282	5968	59
	LR (0.1)	0.521	0.738	0.615	0.394	0.546	0.496	0.242	0.329	0.340	5489	54
tfidf	NB	0.400	0.639	0.482	0.250	0.382	0.333	0.114	0.176	0.173	14	59
	KNN (10)	0.470	0.716	0.563	0.367	0.541	0.467	0.237	0.333	0.336	13,392	3202
	SVM (1)	0.549	0.559	0.630	0.417	0.425	0.508	0.262	0.266	0.351	2631	62
	LR (10)	0.552	0.768^{A}	**0.647**^{A}	0.432^{A}	0.588^{A}	**0.537**^{A}	**0.278**^{A}	0.368^{A}	**0.383**	1856	57
Entropy	NB	0.400	0.638	0.482	0.250	0.382	0.333	0.114	0.176	0.173	141	58
	KNN (5)	0.466	0.693	0.560	0.361	0.519	0.461	0.235	0.318	0.332	13,288	3114
	SVM (1)	0.551	0.560	0.631	0.419	0.426	0.509	0.263	0.268	0.352	3001	60
	LR (10)	0.551	**0.769**^{A}	0.646^{A}	0.431^{A}	**0.589**^{A}	0.536^{A}	**0.278**^{A}	**0.369**^{A}	0.382^{A}	1986	57
w2v	NB	0.111	0.297	0.174	0.040	0.127	0.097	0.010	0.041	0.042	20	54
	KNN (10)	0.373	0.614	0.459	0.270	0.426	0.360	0.164	0.245	0.247	49,648	16,583
	SVM (10)	0.480	0.486	0.555	0.344	0.349	0.426	0.198	0.201	0.273	14,578	92
	LR (100)	0.464	0.705	0.552	0.352	0.521	0.448	0.218	0.311	0.313	4186	82
Only main code												
tf	NB	0.464	0.672	0.528	0.315	0.462	0.381	0.171	0.255	0.224	**10**	**44**
	KNN(10)	0.381	0.577	0.440	0.277	0.405	0.336	0.168	0.244	0.221	4428	1780
	SVM (0.1)	0.487	0.504	0.553	0.348	0.361	0.417	0.204	0.213	0.266	1032	47
	LR (0.1)	0.537	0.736	0.610	0.399	0.539	0.478	0.241	0.329	0.313	1773	45

Table 8 (continued)

Rep.	Classifier	Class			Subclass			Main group			Time (s)	
		Top	Three	All	Top	Three	All	Top	Three	All	Training	Test
tfidf	NB	0.386	0.593	0.460	0.214	0.326	0.278	0.096	0.153	0.139	**10**	45
	KNN (10)	0.501	0.707	0.570	0.387	0.533	0.463	0.247	0.335	0.320	4361	1769
	SVM (1)	0.547	0.555	0.620	0.411	0.418	0.490	0.256	0.260	0.330	747	49
	LR (10)	**0.565**^{A}	0.764^{A}	0.642^{A}	0.434^{A}	0.577	0.520	0.273^{A}	0.360	0.352	954	46
Entropy	NB	0.386	0.593	0.460	0.214	0.326	0.278	0.096	0.153	0.139	149	**44**
	KNN (10)	0.502	0.709	0.571	0.387	0.535	0.464	0.248	0.337	0.321	4622	1745
	SVM(1)	0.548	0.556	0.620	0.412	0.418	0.491	0.256	0.261	0.331	876	49
	LR (10)	0.564^{A}	0.763^{A}	0.640^{A}	**0.434**^{A}	0.577	0.519	0.272^{A}	0.360	0.352	1074	46
w2v	NB	0.119	0.349	0.176	0.058	0.131	0.108	0.019	0.048	0.044	10	50
	KNN (10)	0.412	0.603	0.476	0.293	0.416	0.360	0.175	0.245	0.232	13774	8414
	SVM (10)	0.479	0.485	0.550	0.341	0.347	0.416	0.198	0.203	0.263	6104	75
	LR (100)	0.485	0.691	0.559	0.360	0.504	0.441	0.220	0.305	0.296	1926	61

Table 9 Performance comparison of different document representations and use of all codes or only main code with the WIPO-de dataset

Rep.	Classifier	Class			Subclass			Main group			Time (s)	
		Top	Three	All	Top	Three	All	Top	Three	All	Training	Test
All codes												
tf	NB	0.465^A	0.641	0.528^A	0.343	0.458	0.405	0.174	0.226	0.233	10	41
	KNN (5)	0.275	0.418	0.321	0.204	0.289	0.244	0.127	0.168	0.167	23,862	1721
	SVM (0.1)	0.498^A	0.501	**0.567**^{A}	0.385^A	0.387	0.455^A	0.226^A	0.227	0.300^A	4770	37
	LR (1)	0.490^A	0.660^A	0.557^A	0.385	**0.506**^{A}	**0.456**^{A}	0.229^A	0.294^A	0.305^A	2621	40
tfidf	NB	0.469	0.660^A	0.537	0.345	0.478	0.411	0.178	0.237	0.241	9	39
	KNN (10)	0.425	0.601	0.489	0.339	0.459	0.406	0.212^A	0.275^A	0.282	28,211	1825
	SVM (1)	0.489^A	0.492	0.557	0.374^A	0.376	0.445	0.220^A	0.221	0.292	3040	39
	LR (10)	0.476	0.648	0.546	0.377^A	0.495	0.450^A	0.230^A	0.294^A	**0.308**^{A}	1972	38
Entropy	NB	0.470	0.660^A	0.538	0.345	0.478	0.412	0.178	0.237	0.242	75	40
	KNN (10)	0.426	0.602	0.491	0.341	0.461	0.408	0.213^A	0.276^A	0.283	24,424	2110
	SVM (1)	0.489^A	0.492	0.556	0.374^A	0.376	0.444	0.220^A	0.221	0.292	3068	38
	LR (10)	0.479^A	0.649	0.548	0.378^A	0.495	0.451^A	0.230^A	0.294^A	**0.308**^{A}	2071	40
w2v	NB	0.093	0.260	0.113	0.032	0.100	0.043	0.012	0.038	0.022	20	60
	KNN (10)	0.359	0.567	0.422	0.270	0.405	0.334	0.158	0.223	0.223	189,919	28,256
	SVM (100)	0.453	0.455	0.522	0.331	0.333	0.403	0.182	0.183	0.252	32,748	89
	LR (100)	0.445	**0.666**^{A}	0.521	0.347	0.503^A	0.427	0.210	0.294^A	0.294	8298	70
Only main code												
tf	NB	0.461^A	0.623	0.520^A	0.331	0.432	0.387	0.170	0.216	0.220	7	**30**
	KNN (5)	0.274	0.385	0.311	0.203	0.269	0.236	0.128	0.159	0.161	6477	878
	SVM (1)	0.495^A	0.501	0.557	0.376^A	0.381	0.438	0.223^A	0.226	0.282	1496	29
	LR (10)	0.480	0.630	0.541	0.372^A	0.471	0.436	0.225^A	0.277^A	0.285^A	1005	30

Table 9 (continued)

Rep.	Classifier	Class			Subclass			Main group			Time (s)	
		Top	Three	All	Top	Three	All	Top	Three	All	Training	Test
tfidf	NB	0.470	0.649^{A}	0.531	0.345	0.467	0.404	0.183	0.236	0.236	**6**	30
	KNN (10)	0.429	0.590	0.486	0.340	0.447	0.398	0.213	0.269	0.271	6458	875
	SVM (1)	**0.502**^{A}	0.505	0.566^{A}	**0.387**^{A}	0.390	0.453	0.234^{A}	0.235	0.299	1018	29
	LR (10)	0.490^{A}	0.651	0.552	0.385^{A}	0.495	0.450	0.239^{A}	**0.299**^{A}	0.305^{A}	864	29
Entropy	NB	0.470	0.649^{A}	0.531	0.345	0.467	0.404	0.183	0.237	0.236	86	33
	KNN (10)	0.431	0.590	0.487	0.341	0.448	0.400	0.213	0.269	0.272	7254	982
	SVM (1)	**0.502**^{A}	0.505	0.566^{A}	**0.387**^{A}	0.389	0.453	0.233^{A}	0.235	0.299	1095	30
	LR (10)	0.490^{A}	0.652	0.552	0.385^{A}	0.495	0.450	**0.239**^{A}	**0.299**^{A}	0.305^{A}	963	30
w2v	NB	0.090	0.241	0.115	0.044	0.115	0.057	0.015	0.049	0.026	11	50
	KNN (5)	0.351	0.524	0.403	0.255	0.363	0.307	0.148	0.204	0.197	47,052	13,974
	SVM (100)	0.465	0.467	0.531	0.342	0.344	0.408	0.193	0.195	0.254	12,698	65
	LR (100)	0.450	0.660^{A}	0.518	0.348	0.495	0.418	0.210	0.291	0.281	3271	62

third finding is that LR is a consistent good base classifier with both flat or hierarchical approaches.

Document representations and patent codes

In the second experiment, we compare the effect of using different document representations and using all the codes from patents or only the main code during training. In this case, because of efficiency, we chose to build only hierarchical models. Tables 8 and 9 shows the results of this experiment for WIPO-alpha and WIPO-de respectively. In the case of WIPO-alpha, we use only the abstract section from patents, with WIPO-de we use only the claims section. The first 12 rows of the table correspond to the use of all the codes from patents, and the last 12 to the use of only the main code. Each row corresponds to one base classifier using a given document representation, with the optimal parameter between parentheses, and letter A in super index indicating values that are not significantly different than the best values in a column.

In the results of the second experiment regarding document representations, in both datasets we observe that in general tf–idf and entropy representations produce similar classification results between them, whether using all the codes or only the main code, and better than the ones of tf. Both methods also present generally better results than w2v, except with WIPO-de for some metrics when combining with LR. Models using tf–idf are generally the fastest to be trained, due to entropy requiring more processing time for transforming documents, and base classifiers requiring more computation to adjust models for tf and word2vec. In the case of w2v the training times are much higher than the other representations (except when combining with NB) because of the dense representation. The test times are similar between the same base classifiers regardless of the used document representation; except when using word2vec together with KNN.

Concerning the use of all the codes or only the main code for training, in both datasets we observe that generally the use of all codes produces slightly better classification results, specially at the subclass and main group levels. The better performance at bottom levels of using all the codes is because there are more patents per category at those levels, whilst there is less chance of having an overlap between categories, since patents with several codes are distributed among more categories, and those categories can be far apart in the hierarchy. With respect to the training time, models using only the main code are trained two to five times faster than models trained using all the codes, whilst their test times are similar.

Regarding the base classifier, with the WIPO-alpha dataset the best classification results are from models using LR whether in combination with all the codes or only the main code. With the WIPO-de, it is less clear if there is a dominant base classifier, but LR still keeps performing on top. The more competitive results among base classifiers in this case is because there is more information in the claims section and it helps to build more robust classifiers.

A first finding in this experiment is that using either tf–idf or entropy for document representation is more convenient than tf or word2vec, since they produce better classification results, but tf–idf should be preferred since its training and test processes are faster. A second finding is that using all the codes from patents to train models could produce slightly best results in some cases, but using only the main code produces good results at a fraction of the training time. We could then decide if we want a slightly higher accuracy at a higher

Table 10 Performance comparison of different extracted patent sections/combinations with the WIPO-alpha dataset

Section	Classifier	Class			Subclass			Main group			Time (s)	
		Top	Three	All	Top	Three	All	Top	Three	All	Training	Test
Title	NB	0.460	0.665	0.530	0.306	0.438	0.374	0.166	0.238	0.219	**6**	37
	KNN (10)	0.433	0.613	0.494	0.314	0.427	0.374	0.189	0.250	0.240	1747	532
	SVM (1)	0.494	0.499	0.561	0.350	0.354	0.416	0.206	0.209	0.264	529	38
	LR (10)	0.502	0.690	0.571	0.366	0.493	0.436	0.219	0.294	0.282	510	37
Inventors	NB	0.123	0.238	0.158	0.067	0.112	0.090	0.033	0.052	0.049	**6**	31
	KNN (1)	0.149	0.203	0.171	0.106	0.126	0.124	0.069	0.078	0.086	1214	337
	SVM (1)	0.115	0.116	0.140	0.070	0.070	0.084	0.041	0.041	0.052	513	**27**
	LR (10)	0.132	0.231	0.159	0.086	0.132	0.103	0.053	0.076	0.067	450	31
Abstract	NB	0.464	0.672	0.528	0.315	0.462	0.381	0.171	0.255	0.224	10	44
	KNN (10)	0.501	0.707	0.570	0.387	0.533	0.463	0.247	0.335	0.320	4361	1769
	SVM (1)	0.547	0.555	0.620	0.411	0.418	0.490	0.256	0.260	0.330	747	49
	LR (10)	0.565	0.764	0.642	0.434	0.577	0.520	0.273	0.360	0.352	954	46
Claims	NB	0.451	0.652	0.515	0.320	0.467	0.389	0.182	0.271	0.240	17	50
	KNN (10)	0.554	0.749	0.625	0.439	0.584	0.520	0.289	0.381	0.369	8605	4372
	SVM (1)	0.583	0.592	0.659	0.453	0.460	0.540	0.299	0.304	0.381	1287	71
	LR (10)	0.597	0.788	0.677	0.468	0.613	0.561	0.305	0.395	0.395	1763	65
Desc (S)	NB	0.556	0.746	0.627	0.397	0.545	0.474	0.227	0.314	0.290	18	56
	KNN (10)	0.581	0.778	0.658	0.466	0.610	0.551	0.311	0.401	0.396	8886	5066
	SVM (1)	0.619	0.626	0.698	0.490	0.497	0.577	0.326	0.331	0.412	1519	77
	LR (10)	0.628	0.827	0.712	0.503	0.656	0.597	0.333	0.428	0.425	1937	72
Desc (L)	NB	0.473	0.653	0.543	0.340	0.476	0.414	0.192	0.269	0.252	79	119
	KNN (5)	0.612	0.775	0.687	0.499	0.623	0.586	0.345	0.423	0.436	43,307	30,130
	SVM (1)	0.653^A	0.660	0.732^A	0.529^A	0.535	0.623	0.363^A	0.366	0.458^A	7779	334
	LR (10)	0.653^A	0.838^A	0.735^A	**0.533**^{A}	0.677^A	0.631^A	0.359^A	0.450	0.459^A	10,917	295

Table 10 (continued)

Section	Classifier	Class			Subclass			Main group			Time (s)	
		Top	Three	All	Top	Three	All	Top	Three	All	Training	Test
TIACS	NB	0.517	0.717	0.585	0.380	0.535	0.454	0.221	0.318	0.286	26	64
	KNN (10)	0.612	0.801	0.690	0.504	0.650	0.594	0.345	0.440	0.437	14150	8601
	SVM (1)	0.648	0.656	0.728	0.526^{A}	0.534	0.620	0.360^{A}	0.365	0.455^{A}	1968	106
	LR (10)	0.653^{A}	**0.843**^{A}	**0.739**^{A}	**0.533**^{A}	**0.683**^{A}	**0.633**^{A}	0.361^{A}	**0.457**^{A}	**0.461**^{A}	3098	97
TIACL	NB	0.479	0.651	0.547	0.347	0.480	0.421	0.198	0.276	0.260	87	119
	KNN (5)	0.611	0.776	0.686	0.497	0.623	0.584	0.345	0.423	0.434	43,845	30,137
	SVM (1)	**0.656**^{A}	0.663	0.734^{A}	0.532^{A}	0.537	0.625^{A}	**0.364**^{A}	0.368	0.459^{A}	7770	344
	LR (10)	0.652^{A}	0.840^{A}	0.734^{A}	**0.533**^{A}	0.678^{A}	0.630^{A}	0.360^{A}	0.450^{A}	0.459^{A}	11,634	285

Table 11 Performance comparison of different extracted patent sections/combinations with the WIPO-de dataset

Section	Classifier	Class			Subclass			Main group			Time (s)	
		Top	Three	All	Top	Three	All	Top	Three	All	Training	Test
Title	NB	0.461	0.623	0.520	0.331	0.432	0.387	0.170	0.216	0.220	**7**	30
	KNN (10)	0.429	0.590	0.486	0.340	0.447	0.398	0.213	0.269	0.271	6458	875
	SVM (1)	0.502	0.505	0.566	0.387	0.390	0.453	0.234	0.235	0.299	1018	**29**
	LR (10)	0.490	0.651	0.552	0.385	0.495	0.450	0.239	0.299	0.305	864	**29**
Inventors	NB	0.241	0.406	0.286	0.162	0.254	0.198	0.093	0.139	0.125	11	36
	KNN (5)	0.272	0.400	0.321	0.201	0.274	0.248	0.129	0.167	0.174	6981	946
	SVM (1)	0.261	0.261	0.308	0.183	0.183	0.224	0.111	0.112	0.149	1430	40
	LR (10)	0.270	0.424	0.321	0.196	0.286	0.242	0.127	0.173	0.170	1183	35
Claims	NB	0.458	0.651	0.528	0.340	0.480	0.404	0.156	0.219	0.210	46	50
	KNN (10)	0.534	0.720	0.609	0.447	0.590	0.531	0.300	0.384	0.390	32017	8267
	SVM (1)	0.601	0.606	0.681	0.502	0.505	0.589	0.330	0.333	0.426	3470	63
	LR (C = 100)	0.598	0.777	0.680	0.501	0.642	0.592	0.334	0.421	0.433	4593	60
Desc (S)	NB	0.538	0.709	0.617	0.387	0.510	0.458	0.178	0.228	0.237	32	47
	KNN (10)	0.601	0.792	0.690	0.512	0.665	0.613	0.348	0.441	0.457	26,827	7141
	SVM (1)	0.644	0.650	0.732	0.546	0.551	0.645	0.365	0.368	0.473	2851	55
	LR (100)	0.638	0.820	0.728	0.543	0.692	0.644	0.366	0.459	0.476	3675	54
Desc (L)	NB	0.513	0.698	0.592	0.397	0.540	0.473	0.188	0.253	0.254	162	115
	KNN (10)	0.628	0.813	0.716	0.547	0.696	0.649	0.389	0.482	0.504	105,264	40,055
	SVM (10)	0.680^A	0.687	0.770	0.593^A	0.599	0.697	**0.419**^{A}	0.423	0.540^A	13,896	190
	LR (100)	**0.687**^{A}	**0.863**	**0.780**	**0.599**^{A}	**0.748**	**0.707**	**0.419**^{A}	**0.516**	**0.543**^{A}	19,715	179
TICS	NB	0.529	0.714	0.605	0.405	0.547	0.478	0.190	0.256	0.254	63	65
	KNN (10)	0.605	0.788	0.688	0.522	0.667	0.617	0.366	0.454	0.473	43,414	12,983
	SVM (1)	0.664	0.669	0.751	0.573	0.578	0.671	0.394	0.397	0.504	4824	82
	LR (100)	0.660	0.837	0.749	0.571	0.718	0.673	0.394	0.488	0.509	7171	81

Table 11 (continued)

Section	Classifier	Class			Subclass			Main group			Time (s)	
		Top	Three	All	Top	Three	All	Top	Three	All	Training	Test
TICL	NB	0.515	0.694	0.593	0.405	0.546	0.483	0.197	0.264	0.265	180	119
	KNN (10)	0.620	0.801	0.706	0.539	0.685	0.639	0.384	0.474	0.497	108,530	38,770
	SVM (10)	0.670^{A}	0.677	0.759	0.583	0.589	0.686	0.411	0.416	0.530	17,042	201
	LR (100)	0.678^{A}	0.850	0.768	0.590^{A}	0.735	0.695	0.413	0.506	0.533	22,906	187

computational cost. A third finding is that LR is a consistent good classification model along document representations and use of patent codes.

Using different patent sections

In the third experiment, we measure the effect of using the different patent sections/combinations for extracting information. Because of efficiency, for this experiment we build models using the hierarchical approach, a tf–idf representation and only the main code from patents. Tables 10 and 11 show the results of this experiment for WIPO-alpha and WIPO-de respectively. Each row corresponds to one base classifier using a given extracted section/combination, with the optimal parameter between parentheses and letter A in super index indicating values that are not significantly different than the best values in a column

In this experiment we observe that there exist two general trends in both datasets: the more information a section/combination contains (see Table 5), the better the classification results of the models that use such section/combination; and the more information a section has, the more expensive is to train and test a model, with larger differences observed in training times. There are two exceptions for the first trend: first, using the title section produces better classification results than using the inventors section, and second, using the short description produces better results than using the claims section. We think these two cases are due to two different issues, but both of them related with how the words are distributed over documents and categories. In the first case, an inventor is usually associated with a very small amount of patents usually in the same technological field. Nevertheless, there is the issue of synonymy. Different inventors from different fields share the same names, and since there are many inventors and they could be associated practically with any category, a combination of inventor names as a whole is not a good descriptor of a specific category. In the case of the claims section, it uses a combination of technical words describing the invention and legal words describing the scope of the patent. The legal words are general concepts that are shared by many patents from different categories, and do not carry much information about a specific category. On the contrary, the short description section consists of technical words describing specifically the invention, and contain technical words more associated with specific categories. For these issues, an interesting research direction would be to apply methods that manipulate word features to increase the cohesion between documents from the same category, while at the same time increasing the separability between categories (Gomez and Moens 2010).

Regarding the base classifiers, we also observe for both datasets a general trend in classification results. Independently of the section used, the classifiers sorted in descending order of performance are LR, SVM, KNN and NB; although there are some cases where the order is switched for SVM and LR.

A first finding of this experiment is that, contrary to what other works in the literature have concluded, the full patent description contains relevant and discriminative information to built robust classification models, since using it (or a combination that includes it) produces the best classification results. A second finding is that the short description could be considered a better way to summarize a patent than the abstract section, since the former produces better classification results for all the metrics with all the base classifiers at all the levels. This section should be preferred over abstract when building general patent classification models.

A general finding from all the experiments is that there are not major differences between the models' classification performance regarding the patents' language, and the

Table 12 Performance comparison of most relevant works with results obtained by a model created with our methodology

Work	Structure	Rep.	Codes	Class			Subclass			Main group		
				Top	Three	All	Top	Three	All	Top	Three	All
WIPO-alpha												
1	Flat	tf	Main	0.550	0.790	0.630	0.410	0.620	0.480	–	–	–
2	Hierarchy	Binary	–	0.560	0.810	0.630	0.420	0.670	0.500	–	–	–
4	Hierarchy	Entropy	All	0.655	0.856	0.734	0.532	0.750	0.623	0.368	0.556	0.464
This	Hierarchy	tf–idf	Main	0.653	0.838	0.735	0.533	0.677	0.631	0.359	0.450	0.459
WIPO-de												
3	Flat	tf	Main	0.650	0.860	0.760	0.560	0.780	0.710	–	–	–
4	Hierarchy	Entropy	All	0.650	0.871	0.750	0.554	0.777	0.669	0.380	0.573	0.508
This	Hierarchy	tf–idf	Main	0.687	0.863	0.780	0.599	0.748	0.707	0.419	0.516	0.543

existent differences could be due to WIPO-de having less patents in the test set than WIPO-alpha. This is an indication that classification models are language independent, and their performance is more associated with the distribution of word features over documents and categories. A second general finding is that discriminative classifiers (LR and SVM) tend to perform consistently better than the probabilistic classifier (NB) and the instance-based classifier (KNN). A final finding is that hyper-parameter optimization should be preferred over using the default parameter values to build consistent models.

Comparison with other methods in the literature

Considering the previous findings, we created a model that uses our hierarchical approach, a tf–idf representation, only the main code from patents, LR as the base classifier and the full description to extract the content. This model is fast to train because of the hierarchical approach and the tf–idf representation, includes enough information from the full description to be able to build robust discriminative classifiers, and uses a base classifier that performs consistently good. In this way, it is expected to reach good classification results with a moderate training cost.

In Table 12 we compare the results of our model with the ones obtained by Fall and Benzineb (2002) and Seneviratne et al. (2015) (1 and 2 in table) using WIPO-alpha, Fall et al. (2004) (3 in table) using WIPO-de, and Tikk et al. (2005) (4 in table) using both datasets. In the table we present the best results from the mentioned works.

The results in (1) and (3) were obtained following the same methodology between them: a flat approach, use of stemmed words, a tf representation and only the main category from patents. The authors experimented with a set of base classifiers: multinomial NB, KNN, linear SVM and SNoW in (1), and the same less SNoW plus LLSF in (3); and with different patent sections: title, abstract, claims, full description and the first 300 words of the full description. They did not perform optimization over the parameters of the classifiers and take a value of $K = 30$ for KNN and the value of C is not mentioned. For the SVM classifier they selected a subset of 20,000 words in (1) and 50,000 in (3) using information gain, and limited the amount of document to 500 patents per class. Finally, they trained independent models at the class and subclass levels. In both works, the authors concluded that using the first 300 words from the description is the best option, whilst using the full patent description introduce noise and reduce the models' performance. The results from (1) correspond to different classifiers, at the class level in order of metrics these are: SVM, NB and NB, at subclass level: SVM, KNN and SVM; whilst all the results from (3) correspond to LLSF.

The results in (2) were obtained with a modified KNN method using document signatures with a binary representation and a width of 4096 bits. The authors used a hierarchical approach, and, similar to (1), a value of $K = 30$ and the title or the first 300 words from the description to extract the content. They do not mention if they use all the patent codes or only the main one. They concluded that the first 300 words produce the best results.

The results in (4) were obtained with HITEC (a back-propagation-based model) with the following setup: using a hierarchical approach, stemmed word features and a feature selection based on frequency, eliminating words appearing in less than two patents or in more than 25% of patents. The authors experimented with document representations, combinations of patent sections and use of all the codes or only the main code. The best results were obtained using an entropy representation, all the codes from patents, and a combination of the title, inventors and abstract. An additional conclusion they reached is that using only the main code, the full description and a tf–idf representation produces poor results.

The results obtained with our model are up to 10% better than the ones in (1) and (2) for WIPO-alpha. The improvement come mainly from the selected patent section, (optimized) base classifier and document representation. When comparing with the results in (4), we observe that there is little difference in performance with our model. The largest differences are at the subclass and main group levels with the *three* metric. The difference in performance seems to come from the different base classifiers and patent sections used. Additionally, according to our previous experiments, the flat approach in our model could potentially produce better results, but with an increase in computational cost.

With the WIPO-de dataset the results of our model are between 1% and 4% better than the ones in (3) and (4) for several metrics at several levels, with the exception of the *tree* metric at the subclass and main group levels. The differences seems to come from the selected patent section and base classifier.

Our findings seem to contradict partially what other authors concluded. Our findings indicate that the full description contains relevant and discriminative information for training classification models, the tf–idf representation performs very similar to the entropy, and the use of only the main code could perform similar to using all the codes at some levels of the hierarchy, with the advantage of speeding up the training process.

Our experiments show that analyzing the data properties of the patent classification problem is important to determine the appropriate model depending on our goal, considering classification accuracy and computational efficiency. Some options helps to build models that are slow to train but produce good classification results, while other produce slightly worst results at a fraction of the training time. Thus, the selection of the best options should be chosen following an adequate methodology.

Conclusions

In this paper, we have presented a methodology to conduct a systematic experimental study on automated patent classification, where we analyzed the effect in classification performance of three patent data properties in combination with two general text classification details. The properties we studied were the use of: the different sections to extract content from patents, the different codes assigned to each patent, and the hierarchy of categories. The two details in classification we considered were: the way to represent documents and the base classifier. In our methodology, we trained and tested a diversity of models using different combinations of options for the mentioned properties. We compared the results of the models over two standard patent datasets considering classification accuracy and computational efficiency. Additionally, we showed comparisons with other works in the literature, considering the methodologies, the models and the results.

We could draw several conclusions from our analysis. First, the flat approach produces better results than the hierarchical approach at the lowest level, but it is more expensive to train, specially for discriminative models such as SVM and LR. Second, the use of all the codes assigned to patents could produce better results at the lowest level than using only the main code, but it also increases the training time for a model. Third, the tf–idf and entropy document representations produce similar results, with both producing better results than the tf and w2v representations, whilst tf–idf is computed faster than entropy. Fourth, contrary to what several researchers in previous works claim, the full description is a good source of discriminative information and yield some of the highest results in comparison with other independent sections. The disadvantage of using this section is the time and memory requirements for training a model, because of the large number of features. Nevertheless, the short description could be a good summary of the patent, and even if such section misses some relevant features, this could be alleviated by combining it with other sections, such as the abstract or the claims. Fifth, the discriminative classifiers (SVM and LR) perform better than probabilistic (NB) and instance-based (KNN) classifiers. Sixth, it is important to perform a hyper-parameter optimization to optimize the performance. Finally, the models are language independent, and they depend more on how the words are distributed over documents and categories.

When comparing the results of one model created with our methodology with results from some reference works, we observed some details. First, there is extra confirmation that tf–idf and entropy perform better than tf and w2v. Second, discriminative classifiers (SVM, LR, back-propagation and LLSF) produce the top results and outperform probabilistic and instance based classifiers. Third, the hierarchical approach seems to be a better option than the flat approach up to a certain level of the hierarchy, especially when using the full description and a tf–idf representation. Finally, our model produced generally better, or at least at good, results as the other works. This means that an appropriate choosing of values for the patent data properties is important to obtain a good classification performance, and the best options should be chosen following an appropriate methodology.

It is clear from the results obtained in this work, as well from other works, that the automated classification of patents is still an open problem. The results at the lowest level of the hierarchy are still low to be considered acceptable in a practical setting. Possible research directions include using other features besides word features, such as sentences or topic model representations, in order to include more semantic information from textual content. Some works have already tried using phrases (D'hondt et al. 2013, 2014; Verberne et al. 2010; Verberne and D'hondt 2011), but the performance obtained is similar or even worse than using word features. We thus believe further research is necessary. Another direction could be the study of code propagation between documents that are close related in the hierarchical structure (Rossi et al. 2016). Finally, it would also interesting to study feature selection methods (Lamirel et al. 2015) that find the features that are highly associated with specific categories, maximizing the intra document similarity and minimizing the inter category similarity (Gomez and Moens 2010).

References

Abbas, A., Zhang, L., & Khan, S. U. (2014). A literature review on the state-of-the-art in patent analysis. *World Patent Information*, *37*, 3–13.

Arts, S., Cassiman, B., & Gomez, J. C. (2018). Text matching to measure patent similarity. *Strategic Management Journal*, *39*(1), 62–84.

Basile, A., Dwyer, G., Medvedeva, M., Rawee, J., Haagsma, H., & Nissim, M. (2017). N-gram: New Groningen author-profiling model. arXiv preprint arXiv:1707.03764

Bennett, P. N., & Nguyen, N. (2009). Refined experts: Improving classification in large taxonomies. In *Proceedings of the 32nd international ACM SIGIR conference on research and development in information retrieval* (pp. 11–18). ACM.

Benzineb, K., & Guyot, J. (2011). Automated patent classification. In M. Lupu, K. Mayer, J. Tait, & A. J. Trippe (Eds.), *Current challenges in patent information retrieval* (Vol. 29, pp. 239–261). Berlin: Springer. https://doi.org/10.1007/978-3-642-19231-9_12

Bi, W., & Kwok, J. T. (2014). Mandatory leaf node prediction in hierarchical multilabel classification. *IEEE Transactions on Neural Networks and Learning Systems*, *25*(12), 2275–2287.

Cai, L., & Hofmann, T. (2004). Hierarchical document categorization with support vector machines. In *Proceedings of the 13th ACM international conference on information and knowledge management* (pp. 78–87). ACM.

Chen, Y. L., & Chang, Y. C. (2012). A three-phase method for patent classification. *Information Processing & Management*, *48*(6), 1017–1030.

Cinar, Y. G., Zoghbi, S., & Moens, M. F. (2015). Inferring user interests on social media from text and images. In *2015 IEEE international conference on data mining workshop (ICDMW)* (pp. 1342–1347). IEEE.

Dallachiesa, M., Aggarwal, C., & Palpanas, T. (2014). Node classification in uncertain graphs. In *Proceedings of the 26th international conference on scientific and statistical database management* (pp. 1–4). ACM.

D'hondt, E., Verberne, S., Koster, C., & Boves, L. (2013). Text representations for patent classification. *Computational Linguistics*, *39*(3), 755–775.

D'hondt, E., Verberne, S., Oostdijk, N., Beney, J., Koster, C., & Boves, L. (2014). Dealing with temporal variation in patent categorization. *Information Retrieval*, *17*(5), 520–544.

D'hondt, E., Verberne, S., Oostdijk, N., & Boves, L. (2017). Patent classification on subgroup level using balanced winnow. In *Current challenges in patent information retrieval* (pp. 299–324). Springer.

Fall, C., Törcsvári, A., Fiévet, P., & Karetka, G. (2004). Automated categorization of German-language patent documents. *Expert Systems with Applications*, *26*(2), 269–277.

Fall, C. J., & Benzineb, K. (2002). *Literature survey: Issues to be considered in the automatic classification of patents*. Tech. rep., World Intellectual Property Organization.

Fall, C. J., Törcsvári, A., Benzineb, K., & Karetka, G. (2003). Automated categorization in the international patent classification. *SIGIR Forum*, *37*(1), 10–25.

Fan, R. E., Chang, K. W., Hsieh, C. J., Wang, X. R., & Lin, C. J. (2008). LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, *9*, 1871–1874.

Giachanou, A., & Salampasis, M. (2014). IPC selection using collection selection algorithms. In *Proceedings of the 2014 information retrieval facility conference*, *Lecture Notes in Computer Science* (Vol. 8849, pp. 41–52). Springer

Giachanou, A., Salampasis, M., & Paltoglou, G. (2015). Multilayer source selection as a tool for supporting patent search and classification. *Information Retrieval Journal*, *18*(6), 559–585.

Gomez, J. C., & Moens, M. F. (2010). Using biased discriminant analysis for email filtering. In *Proceedings of the 14th international conference on knowledge-based and intelligent information and engineering systems*, *Lecture Notes in Computer Science* (Vol. 6276, pp. 566–575). Springer.

Gomez, J. C., & Moens, M. F. (2012). Hierarchical classification of web documents by stratified discriminant analysis. In *Proceedings of the 2012 information retrieval facility conference*, *Lecture Notes in Computer Science* (Vol. 7356, pp. 94–108). Springer.

Gomez, J. C., & Moens, M. F. (2014). Minimizer of the reconstruction error for multi-class document categorization. *Expert Systems with Applications*, *41*(3), 861–868.

Gomez, J. C., & Moens, M. F. (2014) A survey of automated hierarchical classification of patents. In *Professional search in the modern world* (pp. 215–249). Springer.

Guyot, J., Benzineb, K., Falquet, G., & Shift, S. (2010). myclass: A mature tool for patent classification. In *Proceedings of CLEF 2010 (notebook papers/LABs/workshops)*.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The weka data mining software: An update. *ACM SIGKDD explorations newsletter*, *11*(1), 10–18.

Härtinger, S., & Clarke, N. (2015). Using patent classification to discover chemical information in a free patent database: Challenges and opportunities. *Journal of Chemical Education*, *93*(3), 534–541.

Iwayama, M., Fujii, A., & Noriko, K. (2005). Overview of classification subtask at NTCIR-5 patent retrieval task. In *Proceedings of the NII test collection for IR systems-5. NTCIR*.

Iwayama, M., Fujii, A., & Noriko, K. (2007). Overview of classification subtask at NTCIR-6 patent retrieval task. In *Proceedings of the NII Test Collection for IR Systems-6. NTCIR*.

Kim, J. H., & Choi, K. S. (2007). Patent document categorization based on semantic structural information. *Information Processing & Management*, *43*(5), 1200–1215.

Koster, C. H. A., Seutter, M., & Beney, J. (2003). Multi-classification of patent applications with Winnow. In *Proceedings of the 5th international Andrei Ershov Memorial*, *Lecture Notes in Computer Science* (Vol. 2890, pp. 546–555). Springer.

Krier, M., & Zaccà, F. (2002). Automatic categorisation applications at the European patent office. *World Patent Information*, *24*(3), 187–196.

Lamirel, J. C., Cuxac, P., Chivukula, A. S., & Hajlaoui, K. (2015). Optimizing text classification through efficient feature selection based on quality metric. *Journal of Intelligent Information Systems*, *45*(3), 379–396.

Li, Y., & Shawe-Taylor, J. (2007). Advanced learning algorithms for cross-language patent retrieval and classification. *Information Processing & Management*, *43*(5), 1183–1199.

Lupu, M., & Hanbury, A. (2013). Patent retrieval. *Foundations and Trends in Information Retrieval*, *7*(1), 1–97.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems* (pp. 3111–3119).

Nanba, H., Fujii, A., Iwayama, M., & Hashimoto, T. (2008). Overview of the patent mining task at the NTCIR-7 workshop. In *Proceedings of the NII test collection for IR systems-7. NTCIR*.

Nanba, H., Fujii, A., Iwayama, M., & Hashimoto, T. (2010) Overview of the patent mining task at the NTCIR-8 workshop. In *Proceedings of the NII test collection for IR systems-8. NTCIR*.

Noh, H., Jo, Y., & Lee, S. (2015). Keyword selection and processing strategy for applying text mining to patent analysis. *Expert Systems with Applications*, *42*(9), 4348–4360.

Piroi, F. (2010). CLEF-IP 2010: Classification task evaluation summary. Tech. Rep. IRF-TR-2010-00005, Information Retrieval Facility.

Piroi, F., Lupu, M., Hanbury, A., & Zenz, V. (2011).CLEF-IP 2011: Retrieval in the intellectual property domain. In *Proceedings of CLEF 2011 (Notebook Papers/Labs/Workshop)*.

Rodriguez-Esteban, R., & Bundschus, M. (2016). Text mining patents for biomedical knowledge. *Drug Discovery Today*, *21*(6), 997–1002.

Rossi, R. G., de Andrade Lopes, A., & Rezende, S. O. (2016). Optimization and label propagation in bipartite heterogeneous networks to improve transductive classification of texts. *Information Processing & Management*, *52*(2), 217–257.

Rousu, J., Saunders, C., Szedmak, S., & Shawe-Taylor, J. (2006). Kernel-based learning of hierarchical multilabel classification models. *Journal of Machine Learning Research*, *7*, 1601–1626.

Seneviratne, D., Geva, S., Zuccon, G., Ferraro, G., Chappell, T., & Meireles, M. (2015). A signature approach to patent classification. In *Proceedings of the 11th Asia information retrieval societies conference*, *Lecture Notes in Computer Science* (Vol. 9460, pp. 413–419). Springer.

Shalaby, W., Zadrozny, W., & Gallagher, S. (2014). Knowledge based dimensionality reduction for technical text mining. In *Proceedings of the 2014 IEEE international conference on big data* (pp. 39–44). IEEE.

Tikk, D., Biró, G., & Yang, J. (2005). Experiment with a hierarchical text categorization method on WIPO patent collections. In *Applied research in uncertainty modeling and analysis*, *International Series in Intelligent Technologies* (Vol. 20, pp. 283–302). Springer.

Trappey, A. J. C., Hsu, F. C., Trappey, C. V., & Lin, C. I. (2006). Development of a patent document classification and search platform using a back-propagation network. *Expert Systems with Applications*, *31*(4), 755–765.

Tsochantaridis, I., Joachims, T., Hofmann, T., & Altun, Y. (2005). Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, *6*, 1453–1484.

Tsoumakas, G., Katakis, I., & Vlahavas, I. (2010). Mining multi-label data. In O. Maimon & L. Rokach (Eds.), *Data mining and knowledge discovery handbook* (pp. 667–685). Boston: Springer. https://doi.org/10.1007/978-0-387-09823-4_34.

Verberne, S., & D'hondt, E. (2011). Patent classification experiments with the Linguistic Classification System LCS in CLEF-IP 2011. In *Proceedings of CLEF 2011 (Notebook Papers/Labs/Workshop)*.

Verberne, S., Vogel, M., & D'hondt, E. (2010). Patent classification experiments with the linguistic classification system LCS. In *Proceedings of CLEF 2010 (Notebook Papers/LABs/Workshops)*.

Wang, D., Ferraro, G., Suominen, H., & Jefferson, O. A. (2014). Automated categorisation of patent claims that reference human genome sequences. In *Proceedings of the 2014 Australasian document computing symposium* (pp. 117–120). ACM.

Wang, X. L., Chen, Y. Y., Zhao, H., & Lu, B. L. (2014). Parallelized extreme learning machine ensemble based on min-max modular network. *Neurocomputing*, *128*, 31–41.

Wang, X. L., Zhao, H., & Lu, Bl. (2014). A meta-top-down method for large-scale hierarchical classification. *IEEE Transactions on Knowledge and Data Engineering*, *26*(3), 500–513.

Zhang, L., Li, L., & Li, T. (2015). Patent mining: A survey. *ACM SIGKDD Explorations Newsletter*, *16*(2), 1–19.

Zhang, X. (2014). Interactive patent classification based on multi-classifier fusion and active learning. *Neurocomputing*, *127*, 200–205.

Zhu, F., Wang, X., Zhu, D., & Liu, Y. (2015). A supervised requirement-oriented patent classification scheme based on the combination of metadata and citation information. *International Journal of Computational Intelligence Systems*, *8*(3), 502–516.