A Survey of Automated Hierarchical Classification of Patents

Juan Carlos Gomez and Marie-Francine Moens

KU Leuven, Department of Computer Science
Celestijnenlaan 200A, 3001 Heverlee, Belgium
{juancarlos.gomez,sien.moens}@cs.kuleuven.be

Abstract. In this era of “big data”, hundreds or even thousands of patent applications arrive every day to patent offices around the world. One of the first tasks of the professional analysts in patent offices is to assign classification codes to those patents based on their content. Such classification codes are usually organized in hierarchical structures of concepts. Traditionally the classification task has been done manually by professional experts. However, given the large amount of documents, the patent professionals are becoming overwhelmed. If we add that the hierarchical structures of classification are very complex (containing thousands of categories), reliable, fast and scalable methods and algorithms are needed to help the experts in patent classification tasks. This chapter describes, analyzes and reviews systems that, based on the textual content of patents, automatically classify such patents into a hierarchy of categories. This chapter focuses specially in the patent classification task applied for the International Patent Classification (IPC) hierarchy. The IPC is the most used classification structure to organize patents, it is world-wide recognized, and several other structures use or are based on it to ensure office inter-operability.

Keywords: hierarchical classification, patent classification, IPC, WIPO, patent content, text mining.

1 Introduction

When a new patent application arrives at the office of one of the organizations in charge of issuing patents around the world, one of the first tasks is to assign classification codes to it based on its content. In this way, it is ensured that patents and patent applications with similar characteristics, dealing with similar topics or in specific technological areas are grouped under the same codes. Accurate classification of patent documents (or simply *patents*, referring to granted patents or patent applications) is vital for the inter-operability between different patent offices and for conducting reliable patent search, management and retrieval tasks, during a patent application procedure. These tasks are crucial to companies, inventors, patent-granting authorities, governments, research and development units, and all individuals and organizations involved in the application or development of technology.

G. Paltoglou et al. (Eds.): Professional Search in the Modern World, LNCS 8830, pp. 215–249, 2014.

However, the more patents there are, the more complex the classification process becomes. This is observed mainly in two directions: first, when there are many patents to manage, the classification structure should be very well organized and detailed to allow easy classification, navigation and precise search. Moreover, since patents somehow reflect the technological knowledge of the world and this knowledge changes over time, the classification structure should also be flexible enough to capture such changes. One valuable approach to deal with the previous details is to use hierarchies of concepts, where the more general concepts or subjects are at the top levels and the more specific ones at the lower levels. The most important structures to organize patents, like the International Patent Classification (IPC), follow such an approach. Second, when a great amount of patents arrive to be processed in a patent office, they need to be classified in the hierarchical structure in a short period of time. Traditionally this has been done manually by patent experts. Nevertheless, in this era of "big data", where a large amount of data in many forms are generated every day, hundreds or even thousands of patent applications arrive daily to patent offices around the world, and the professional experts are becoming overwhelmed by these great amounts of documents. For example, the number of patent applications received by the United States Patent and Trademark Office (USPTO) in 2000 amounted to 380,000, reaching approximately 580,000 in 2012 [66]. The European Patent Office (EPO) received approximately 180,000 patent applications in 2004; this number increased to 257,000 in 2012 [18]. If we add that the hierarchical structures of classification are very complex (containing thousands of concepts/categories) and that experts are costly and vary in capabilities, reliable, fast and scalable methods and algorithms are needed in order to help the experts in the patent classification tasks and to automatize part of the classification process.

This chapter is meant to describe, analyze and review the building of systems that, based on the content of patents, automatically classify patents into a hierarchy of categories. We call this task automated hierarchical classification of patents (AHCP).

The content in a patent is well-structured (divided by sections and fields) and composed of text, figures, draws, plots, etc. Every component of a patent provides useful information to conduct the classification. In this chapter we focus only on the textual content, since it is one of the largest components in patents and several other elements in the content are usually explained using phrases, concepts or words. It is then possible to mention that the AHCP is an instance of the more general hierarchical text classification (HTC) task.

This chapter describes the AHCP as a task of HTC applied particularly for the International Patent Classification (IPC) hierarchy (or simply *IPC*). We use the IPC hierarchy since it is the most used classification structure to organize patents in the world. Other classification structures, such as the European CLAssification (ECLA), the Japanese File Index (FI) and the new Cooperative Patent Classification (CPC), were designed taking the IPC as a basis; while the United States Patent Classification (USPC) uses the IPC codes to maintain communication with other offices. Furthermore, most of the systems for AHCP in the IPC could be extended to other hierarchical structures, since the most used hierarchies follow the same structural and organizational principles as the IPC (not the same categories, but the way they are organized).

Patent classification is closely related to patent search, which is a professional search task. Patent classification and search are tasks conducted by experts in patent offices and other patent-related organizations around the world. Patent classification could be seen by itself as a search task, where the goal is to find and assign the most relevant category codes for a given patent. Assigning the most appropriate codes for a patent is a fundamental step in several tasks of patent analysis. For example, in prior art search, the assigned categories could help to narrow the search when looking for relevant patents. Moreover, the category codes assigned to a patent are language independent, which facilitate retrieval tasks in multi-language environments.

This chapter is very relevant to the objectives of the EU-funded COST Action MUMIA. First, it relates with the working group of Semantic Search, Faceted Search and Visualization in terms of the automatic hierarchical classification of patents based on their content. Faceted classification allows the assignment of multiple classifications to an object, enabling the classifications to be ordered in multiple ways. Faceted search could then rely on several hierarchical structures at the same time, where those structures can reflect different properties of the patent content. This relates our chapter with the fourth secondary objective defined in the Memorandum of Understanding (MoU) of the MUMIA COST Action: To critically examine the use of Taxonomies for Faceted search. Second, the contribution of this chapter consists on providing a survey of works devoted to the AHCP in the IPC. The survey offers an overview of existing technologies and pinpoints their shortcomings. This study could provide to other researches with valuable information about the relevant current methods for AHCP and the research questions still open in the subject. This should encourage further research work for the AHCP. This correlates with the main objective of the MUMIA COST Action, defined in its MoU, by fostering research in areas related with multi-lingual information retrieval, given that patent is by nature a multi-lingual domain and that the AHCP is a relevant task for patent search and retrieval in large-scale digital scenarios.

The rest of this chapter is organized as follows: the IPC is described in section 2. The particularities of the AHCP in the IPC are given in section 3, including the constraints in classification for this task, the structure of patents and the distribution of patents in collections. Section 4 presents the formal definition of hierarchical text classification, the several components that could be used in an AHCP system, and review several recent works focused on tackling the AHCP in the IPC. In section 5 we present our conclusions and various possibilities and perspectives in the near future for AHCP.

2 International Patent Classification

There exist several classification structures (proposed by the different patent offices around the world) to organize patents. The most recognized ones are the

European CLAssification (ECLA), used by the European Patent Office (EPO), the United States Patent Classification (USPC), proposed by the United States Patent and Trademark Office (USPTO), the Japanese F-Terms and the Japanese File Index (FI), devised by the Japanese Patent Office (JPO), and the International Patent Classification (IPC), used internationally. In addition, recently the EPO and the USPTO launched a project to create the Cooperative Patent Classification (CPC) in order to harmonise the patent classifications between the two offices [12]. Among the previous structures, the IPC is considered as the most widely spread and globally agreed. Some other structures, such as the ECLA, FI and the new CPC, are based on it, and others (like the USPTO) use it for helping maintaining a communication with other offices.

The IPC was created under the Strasbourg Agreement in 1971 and it is administered and maintained by the World Intellectual Property Organization (WIPO) [73]. The IPC is used in a worldwide context, having 95% of all existing patents classified according to it and used in more than 100 countries. The IPC is updated periodically by groups of experts, and until 2005 this updating was done every five years. Currently the IPC is under continual revision, with new editions coming into force on the 1st of January each year. The current version is IPC2014.01.

Every category in the IPC is indicated by a code and has a title [72][73]. The IPC divides all technological fields into eight sections designated by one of the capital letters A to H. Each section is subdivided into classes, whose codes consist of the section code followed by a two-digit number, such as B64. Each class is divided into several subclasses, whose codes consist of the class code followed by a capital letter, for example B64C. Each subclass is broken down into main groups, whose codes consist of the subclass code followed by a one- to three-digit number, an oblique stroke and the number 00, for example B64C 25/00. 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, for example B64C 25/02. Subgroups are ordered in the scheme as if their numbers were decimals of the number before the oblique stroke. For example, 3/036 is to be found after 3/03 and before 3/04, and 3/0971 is to be found after 3/097 and before 3/098. The hierarchy after subgroup level is determined solely by the number of dots preceding their titles, i.e. their level of indentation, and not by the numbering of the subgroups.

An example of a sequence of category codes along the different levels of the IPC is shown in table 1 (extracted from [72]). The IPC has then 5 levels in its hierarchy: sections, classes, subclasses, main groups and subgroups. The total number of categories per level of the IPC is shown in table 2.

2.1 Graphical Description of the IPC

The IPC structure could be considered as a rooted tree graph, which in turn is a kind of directed acyclic graph (DAG). In the rooted tree, every category is represented as a vertex or node in the graph. The hierarchy has a root node from where the rest of the nodes depart. The nodes are connected by directed edges

Table 1. Example of a sequence of codes along the different levels of the IPC

IPC	Code	Title
Section	B	Performing operations; Transporting
Class	B64	Aircraft; Aviation; Cosmonautics
Subclass	B64C	Aeroplanes; Helicopters
Main group	B64C 25/00	Alighting gear
Subgroup	B64C 25/02	Undercarriages

Table 2. Number of categories in each level of the IPC

Level	Name	No. of Categories
1	Section	8
2	Class	129
3	Subclass	638
4	Main Group	7391
5	Subgroup	64046

which represent PARENT-OF relationships (with the parent at the beginning of the edge and the child at the end), and every node can only have one parent node, i.e. any node can only have exactly one simple path from the root to it. In the IPC the parent nodes represent more general concepts than the child nodes. The lowest nodes of the tree are named *leaf* nodes. Figure 1 shows a portion of the IPC hierarchy representing the tree graph. As mentioned above, the root node is considered as level 0 of the IPC.

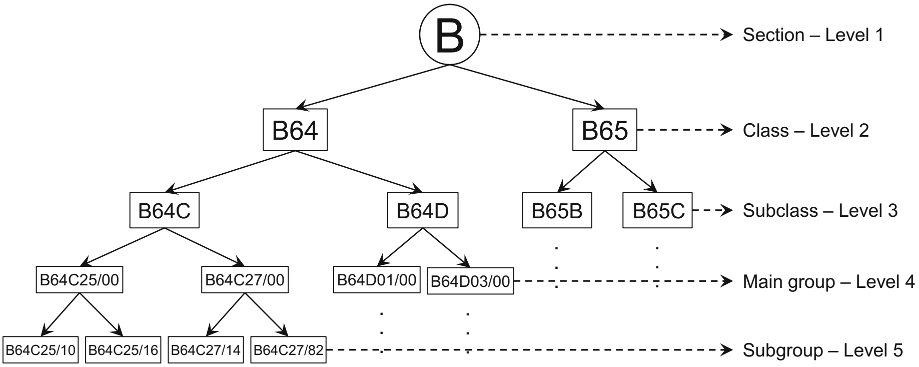

Fig. 1. Example of a portion of the IPC hierarchy starting in level 1, section B. The root node is level 0 (not shown).

Following the definitions of Silla and Freitas [55] and Wu et al. [75], we can say that the IPC is a rooted tree hierarchy Υ defined over a partial order set $(\mathbb{C}, \prec)$, where $\mathbb{C} = \{c_1, c_2, \ldots, c_p\}$ is the previously defined set of possible categories over Υ, and $\prec$ represent the PARENT-OF relationship, which is asymmetric, anti-reflexive and transitive. We then have:

- The origin of the graph is the root of the tree
- $\forall c_i, c_j \in \mathbb{C}$, if $c_i \prec c_j$ then $c_j \not\prec c_i$
- $\forall c_i \in \mathbb{C}$, $c_i \not\prec c_i$
- $\forall c_i, c_j, c_k \in \mathbb{C}$, if $c_i \prec c_j$ and $c_j \prec c_k$ then $c_i \prec c_k$

Up to the main group level, the IPC category codes indicate by themselves paths in the hierarchy. That is, the codes are aggregations of the codes from the root until a given level (with the exception of the root that is never included in the codes). However, at the subgroup level the IPC uses a different way to assign the codes. It uses a dot indentation system. The number of dots indicate the level of the hierarchy for a given code. At the subgroup level is not possible to look at the code and define directly a path in the hierarchy.

Usually, the codes in the leaf nodes of the IPC are the ones assigned to a patent. This would correspond to the codes of the subgroup level. However, if there exist some restrictions, it is also possible to assign a code only up to a certain level of the IPC. One of such restrictions is given by the WIPO itself, where they specify that industrial property offices that do not have sufficient expertise for classifying to a detailed level have the option to classify in main groups only (level 4 of the IPC) [73].

3 Details of the AHCP in the IPC

The general features of the AHCP in the IPC are the following: first, it is hierarchical, since the categories to be assigned follow hierarchical dependencies, where each category is a specialization of some other more general one. Second, it is multi-label, since each patent could have several categories assigned at the same time, i.e. the categories are not mutually exclusive and some could even be correlated. Indeed, the number of possible categories to be assigned to a patent could range from just a few to thousands depending on the area or subarea where the patent must be classified and the level of the hierarchy. Third, it could be partial, since the classification could be conducted only up to a certain level of the hierarchy, depending on the restrictions imposed by the expert users (or by other external factors).

The multi-label issue is a complex one. Firstly, there is not a limit for the number of categories a patent can be assigned, so in principle a patent could have an unlimited number of categories. During the test phase of any given AHCP system, this is an important issue, since the system could output from one to thousands of categories, influencing its performance. Secondly, since a patent in the training data belongs to more than one category, how to consider to which category it belongs when building a classification model is an important issue that also has influence on the performance of the AHCP system [34]. For example, in the collection of patents from the WIPO-alpha dataset [72][1] the maximum number of assigned categories to a patent is 25 and the average number is 1.88 with a standard deviation of 1.43. In the collection of patents from the CLEF-IP 2011 dataset the maximum number of assigned categories to a patent is 102 and the average is 2.16 with a standard deviation of 1.68.

Because of this multi-label issue, the AHCP in the IPC is considered as well as a task where high recall is preferred. That means that recall is an important aspect to consider when developing a system and when evaluating it. A high recall means that it is usually more important to assign the patent to many categories, rather to miss a relevant category. When conducting patent analysis, missing a relevant category for a patent could produce poor search results and in consequence it could lead to legal and economical complications because of patent infringement.

Nevertheless, high recall usually comes at the expense of low precision (several of the categories assigned by a system to a patent could not be relevant for the patent). Because of that, it is usually an important factor for an AHCP system to consider a confidence level when assigning a category for a patent [35]. Using a level of confidence could help to avoid the hurting in performance regarding precision by only allowing the assigning of categories for which the system is really confident. This would also save time to the expert users when analyzing the output of the system.

In order to better define the AHCP in the IPC, we use and extend here the notation by Silla and Freitas [55]. We can then describe the AHCP in the IPC as a 3-tuple $< T, ML, PD >$, where T specifies that the hierarchy Υ used in the task (the IPC) is defined as a rooted tree; ML that the task is multi-label (i.e. several categories could be assigned to a patent) and PD (standing for partial depth) that the task could be conducted only up to a certain level of the hierarchy (depending on the restrictions defined by the expert users in charge of the system or other external restrictions).

The AHCP in the IPC is indeed a complex task, given the large number of categories in the IPC, the variable number of possible categories in each subarea and given that there is not a fixed or specific number of categories to be assigned to a patent.

In addition to the characteristics of the AHCP as a general task, there are other issues that have an influence on the task. These issues are described in the following two subsections.

3.1 Patent Structure

Patents are complex documents and present some differences w.r.t other documents that are usually automatically classified (like news, emails or web pages): patents are long documents (up to several pages), their content is governed by legal agreements and is therefore well-structured (divided by sections and usually

[1] The WIPO-alpha dataset and the CLEF-IP 2011 dataset will be used in the following sections to illustrate the several issues regarding the AHCP in the IPC, and will be explained with more detail in section 4.6.

with well defined paragraphs) and they use natural language in a formal way, with many technical words and sometimes fuzzy sentences (in order to avoid direct similarities with other patents and to extend the scope of the invention).

The structure of a patent is important because it allows to provide different types of input data to an AHPC system; which directly influences the performance of the system during training and testing. Although there are several ways to represent the structure of a patent (with more or less details and different ways of grouping the information), the content of most patents is organized in the following way [4][40][72].

- **Title:** indicates a descriptive *name* of the patent.
- **Bibliographical data:** contains the ID number of the patent, the names of the inventor and the applicant, and 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 which application fields the patent is sought for.

In addition to the previous fields, it is also frequent to find graphics, plots, draws or other types of figures. Every component of a patent provides useful information to conduct the classification. In this chapter we focus only on the textual content, since it is usually one of the largest components in patents and several other elements in the content are often explained using phrases, concepts or words.

The several sections of a patent are usually presented in a XML format. Figure 2 presents an example of the XML structure of a patent extracted from the WIPO-alpha dataset [72].

The sections of a patent vary largely in size, with the title usually being the shortest section and the description the longest. To illustrate this, table 3 presents the number of words appearing in the collections of patents from the WIPO-alpha dataset and the CLEF-IP 2011 dataset. The table shows the minimum, maximum and average number of words per section, counting them in two ways: total words (counts every word in the patent, even if it is a repeated word) and unique words (if a word appears more than once in a patent it only counts as one). The words counted do not include stop words and words composed of less than 3 characters. We observe in this table that the description is by far the longest section, the second is the one containing the claims, the third is the abstract and the shortest one is the title. We also can see that the averages of total and unique words in both datasets are similar.

As mentioned above, the use of the different sections of a patent in the AHCP task is an important issue, since the amount and quality of data processed by a system affects its performance in terms of computing or processing time (efficiency), and in terms of the results it presents to the user (efficacy). Which section, portion,

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE record SYSTEM "../../../../ipctraining.dtd">
<record cy="WO" an="AU9700792" pn="WO992646519990603" dnum="9926465" kind="A1">
<ipcs ed="6" mc="A01B00116">
<ipc ic="A01M02100"></ipc>
</ipcs>
<pas>
<pa>ANDERSON, Frank, Malcolm</pa>
</pas>
<tis>
<ti xml:lang="EN">HYDRAULIC PROBE FOR PLANT REMOVAL
</ti>
</tis>
<abs>
<ab xml:lang="EN">A movable device to facilitate removal of plants with roots intact
from a soil or growing medium is disclosed. The device comprises a rigid
hollow shaft
[... abridged ...]</ab>
</abs>
<cls>
<cl xml:lang="EN">CLAIMS
The claims defining the invention are as follows:1. A movable device facilitating plant
removal with roots intact from a soil or growing medium, the device comprising a rigid
hollow shaft with one end
[... abridged ...]</cl>
</cls>
<txts>
<txt xml:lang="EN"> HYDRAULIC PROBE FOR PLANT REMOVAL
DESCRIPTION
This invention relates to a device for aiding the removal of individual plants with roots
intact from a soil or growing medium.There are several methods for removing plants from
a soil or growing medium.
[... abridged ...]</txt>
</txts>
</record>
```

Fig. 2. Example of the XML structure of an abridged patent from the WIPO-alpha dataset

Table 3. Statistics on number of words in each section of the WIPO-alpha and CLEF-IP 2011 patent datasets

Section	WIPO-alpha						CLEF-IP 2011					
	Total Words			Unique Words			Total Words			Unique Words		
	Min	Max	Average	Min	Max	Average	Min	Max	Average	Min	Max	Average
Title	1	33	5.4	1	23	5.2	1	111	10.3	1	36	5.6
Abstract	2	277	58.5	2	146	36.1	2	1407	67.4	2	625	37.7
Description	63	354769	3072.8	40	86337	747.3	8	1290673	3107.2	8	302867	656.7
Claims	5	32507	539.5	5	13737	103.8	2	89746	447.8	2	11339	121.2

or combination of sections is the best to provide useful information for the AHCP task is still an open question, as we will discuss in section 4.7.

3.2 Other Issues for the AHCP in the IPC

In addition to the generalities of the AHCP in the IPC and the structured content of the patents, there are other issues that have an influence on the task.

The first issue is related to the distribution of patents along the predefined categories of the IPC. The IPC is an artificially created structure that is defined by human experts. As a consequence it imposes external criteria to classify patents, instead of following a definition of the categories based on the "natural" content of patents. In addition, since the focus of research and technological development changes over time, so do the categories in the IPC. These two previous details affect the categories of the IPC in two ways: some categories receive many patents in a given point of time, and the IPC structure changes over time, including the creation and merging (because of deprecation) of categories. This variability in turn creates a highly imbalanced distribution of patents across the IPC. They tend to follow a Pareto-like distribution, with about 80% of them classified in about 20% of the categories [4][19]. To illustrate this effect, figures 3.a and 3.b show the distribution of patents across the categories present in the WIPO-alpha dataset and the CLEF-IP dataset respectively. The categories extracted correspond to the main group level in the IPC. The plots show the number of categories containing between 1 to 50 patents, 51 to 100, and so on. For the WIPO-alpha dataset, we see in the figure that of a total of 5,907 categories, around 89% (5,260) contain only between 1 to 50 patents, while only around 0.02% (1) contain more than 2,000 patents. For the CLEF-IP 2011 dataset, we see that of a total of 7,069 categories, around 28% (1,991) contain only between 1 to 50 patents, while only around 8% (550) contain more than 2,000 patents.

The second issue is related with the previous mentioned details of the dynamical nature of the IPC [19]. This dynamics implies the creation and deprecation (or merge) of categories over time, which in turn affects the performance of an AHCP system, since the definitions of categories could be modified in a given moment, and part of the system could be outdated to classify some patents.

The third issue is related with the distribution of words inside the patents. As seen in the previous section, a patent can contain up to thousands of words. However, of these words only a small portion corresponds to unique words in each patent; and moreover, most of the words appearing in a collection of patents are used very rarely (they are only mentioned in a couple of patents). Similarly than in collections of other documents [38], the distribution of words in a collection of patents tend to follow approximately Zipf's law [4]. To illustrate this fact, figures 3.c and 3.d show the frequency of words in the collection of patents from the WIPO-alpha dataset and the CLEF-IP 2011 dataset. The figures show how many words appear in only 2, 3, 4 and so on patents. The words extracted form the collection do not include stop words, words composed of less than 3 characters and ignores those that are used in only 1 patent. For the WIPO-alpha dataset we observe that from the total vocabulary of 480,422 words, 189,402 words (corresponding to almost 40% of the total) appear in only 2 patents, while 103,607 words (corresponding to around 22% of the total) appear in more than 10 patents. For the CLEF-IP 2011 dataset we observe that from the total vocabulary of 7,373,151 words, 2,685,340 words (corresponding to around 36% of the total) appear in only 2 patents, while 1,424,050 words (corresponding to around 19% of the total) appear in more than 10 patents.

The two mentioned issues of scarcity (lack of data) in most of the categories and the fact that most of the words in a collection of patents are infrequent, largely affect the performance of an AHCP system. To train robust classification

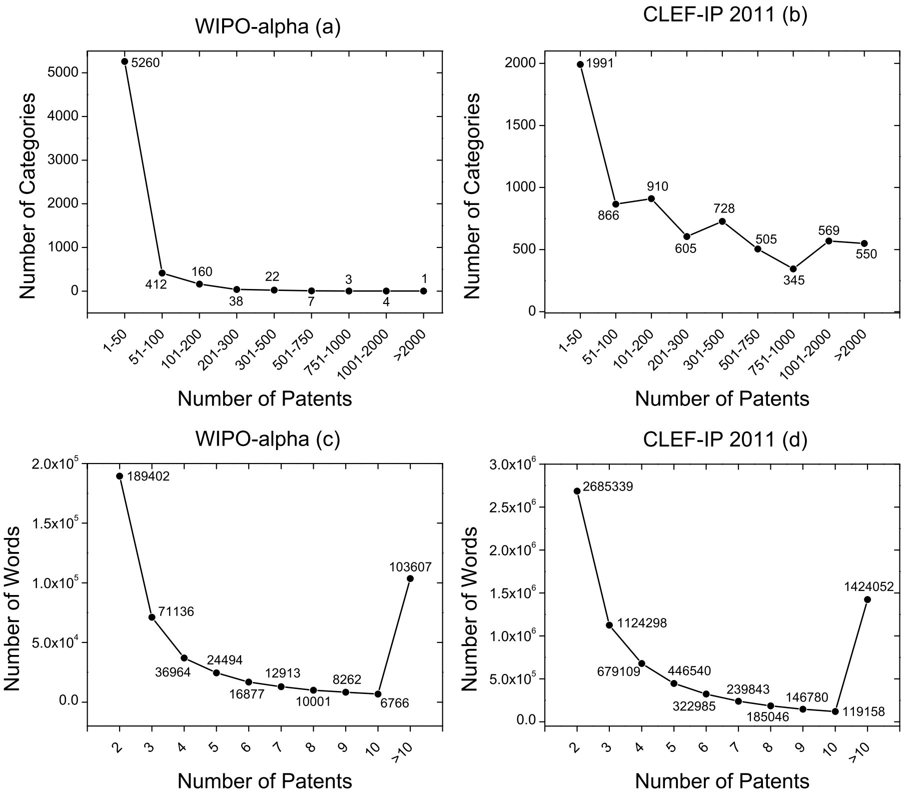

Fig. 3. Statistics in the collections of patents from the WIPO-alpha dataset and the CLEF-IP dataset. (a) and (b) number of patents per category. (c) and (d) frequency of words.

models, a sufficient amount of training data is required [3]. In addition, most of the words are rare, but since most of the categories are rare as well (by the number of patents it contains), it means that some rare words are descriptive of some rare categories and should be kept; imposing the use of a large number of words in the system. This could lead to the so called *curse of dimensionality* [5] for some classification methods.

The fourth issue is related to the citations (or links) inside the patents. Patents are linked to other patents and documents by references to prior art or examples of similar technology. The links could have an effect on the performance of an AHCP system, since usually patents are linked with other patents in the same categories. However, this is still not completely clear, as we will see in section 4.7.

The final issue is related with the language of the patents. By its nature the AHCP in the IPC is a multi-lingual and cross-lingual task. As a matter of generality it should be possible to automatically classify any patent written in (almost) any language by the IPC codes [40]. This is indeed a very complex and hard issue for the AHCP. In order to build models in different languages it is necessary to have training data in such languages; however to acquire such data is not so trivial. That would imply to train a model using patents written in one language and use it with patents in other languages. Furthermore, the use of different languages in patent collections imposes by itself some issues regarding the linguistical particularities of each language, such as [4]: polysemy, synonymy, inflections, agglutination (some languages like German and Dutch stick together several words to build a new word), segmentation (choosing the correct number of ideograms which constitute a word in Asian languages), etc.

Table 4 summarizes the discussed issues regarding the AHCP in the IPC.

Table 4. Summary of the several issues related with the AHCP in the IPC

Issue	Description
Hierarchical	The categories are structured following hierarchical dependencies.
Multi-label	One patent can have more than one category assigned. However, there is not a fixed number of categories to be assigned to each patent.
Partial-depth	The classification could be stopped in any level of the hierarchy.
Patent structure	Patents are structured and composed of several sections.
Distribution of patents in the categories	Most of the patents are distributed in only a few categories.
Distribution of words inside the patents	Most of the words in a collection of patents are very rare, appearing in only a few patents.
Citations	Patents are related with other patents and documents by references.
Language	Patents are written in many languages. Each language needs training patents and imposes linguistical particularities to the task.

4 Recent Models and Advances for the AHCP in the IPC

There are two main points of view for models applied to the AHCP: the first one involves people working with patents and whose main interest is to develop a complete system to assist the experts in the classification of the patents [36][35][56][70]. The second point of view involves the data mining/machine learning communities, where they aim to develop efficient methods to perform the classification task [1][64][50][69]. The first approach uses the methods from the second to accomplish their task, but they put more emphasis on the usability of the final tools and not on the high performance of the methods. The second approach focuses on understanding the structure of the patent data and then tries to derive efficient and effective methods to conduct the classification. Both approaches converge and merge sometimes in the literature; however there still seems to exist a communication gap between the two.

This section presents a revision of several works for the AHCP in the IPC. The works revisited here come from literature in areas related to the two points of view mentioned above. Our goal is to produce a normalized and structured analysis of the works; using for that a defined set of components.

In the direction of structuring our analysis and with the intention of better understanding the AHCP in the IPC, we give first in the next subsection a more formal definition of the general hierarchical text classification (HTC) task, from where the AHCP is derived. Later, we see also the components that could be included in an AHCP system and we describe the possible approaches to reach the goal of AHCP.

4.1 Hierarchical Text Classification

The HTC is divided in two phases: training and testing. For training we have a hierarchical structure $\varUpsilon$ that is composed by a set $\mathbb{C} = \{c_1, c_2, \ldots, c_p\}$ of possible categories that follow the restrictions imposed by the hierarchy. We also have a set of n previously classified text documents $\mathbb{X} = \{(\mathbf{d}_1, \zeta_1), \ldots, (\mathbf{d}_n, \zeta_n)\}$; where $\mathbf{D} = \{\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_m\}$ is the training document matrix, with $\mathbf{d}_i \in \mathbb{R}^m$ as the i-th document represented by a m dimensional column vector; and $\mathbf{L} = \{\zeta_1, \zeta_2, \ldots, \zeta_n\}$ is the category matrix, with $\zeta_i \subset \mathbb{C}$ as the set of categories assigned to document $\mathbf{d}_i$. The objective of the training phase is to build a classification model $\varOmega$ over the hierarchical structure $\varUpsilon$ using the previously classified documents $\mathbb{X}$.

In this definition, the model $\varOmega$ is understood as a black box. Inside it there could be several components, phases or steps, such as base classifiers, meta classifiers, hierarchical management processes, etc. There are many ways of building $\varOmega$, using different components, as we will see later.

For testing we have the hierarchical trained model $\varOmega$ and a set of k unclassified documents $\mathbf{U} = \{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_k\}$, with $\mathbf{u}_i \in \mathbb{R}^m$. The objective in this phase is then to use the model $\varOmega$ to predict or assign a set $\mathbf{V} = \{\nu_1, \nu_2, \ldots, \nu_k\}$ of valid categories to each document $\mathbf{u}_i$. $\mathbf{V}$ is the resulting category matrix for the test documents, with $\nu_i \subset \mathbb{C}$ as the set of assigned categories to $\mathbf{u}_i$. The model $\varOmega$ and the assigned categories $\mathbf{V}$ implicitly follow the restrictions imposed by the hierarchy $\varUpsilon$.

The AHCP in the IPC is indeed an instance of the HTC task. The goal of the ACHP in the IPC is to assign a set of category codes to a given patent, considering the particularities of the IPC hierarchy and the issues of the patent data and the task itself, as seen in sections 2 and 3. The classification model $\varOmega$ from the above definition represents any AHCP system.

4.2 Steps and Components of an AHCP system

Figure 4 shows a general schema of a system performing the AHCP in the IPC [63][19]. The schema is divided in several stages. The process starts with a collection of patents assuming they are in an electronic readable format. The first stage consists of cleaning the collection by eliminating noisy patents (patents that are not electronically readable) and standardizing them to a given format (for example using XML to define the sections). The second stage is the preprocessing of the

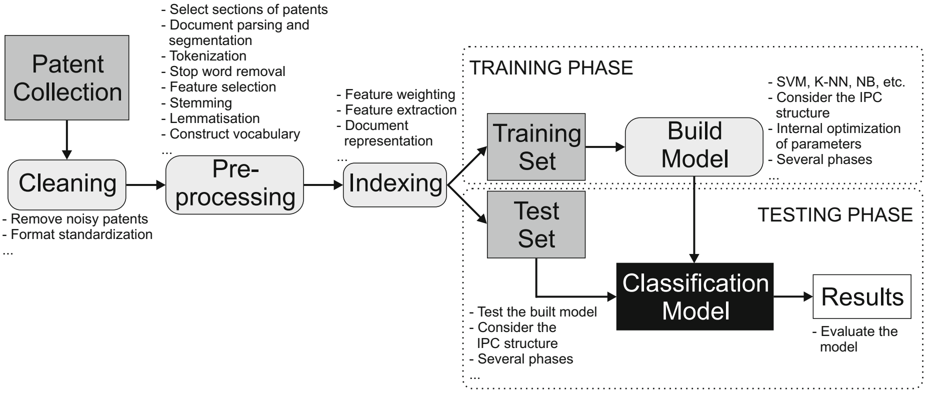

Fig. 4. General steps in the AHCP

patents. This stage could consist of several steps such as: selection of patent sections, tokenization (breaking the text into words, n-grams, phrases, paragraphs, etc. which are called *features*) [71], stop word removal, feature selection (removing the features that are less relevant for the classification task) [78][23], stemming or lemmatisation (grouping together the different inflected forms of a word) [32], vocabulary construction (indexing the features), etc. The third stage is indexing the patent. This stage also could include several steps, such as: feature weighting (how important is each feature for a patent/category), feature extraction (constructing new features using combinations of the original ones) [24], document representation (representing the patents in a format that an algorithm can understand, like vectors, matrices, lists, maps, etc.), among others. Once the patents are processed and expressed in a format that is understandable for a computer, they are divided in a training set and a test set. The training set is used to build the AHCP system, while the test set is held out apart to test the performance of the system. Then, there are two later phases in the process, the training and the testing. During training, as specified in subsection 4.1, the objective is to build a model Ω (understood as the AHCP system) using the already classified set of training patents. The training phase could be done in several steps depending on what base classification algorithms are used (like the optimization of the meta parameters of some of them), how the IPC is used to build the model or if the training is done in several phases, among others. The testing phase consists of providing a set of unclassified patents to the system and obtain a set of categories for each of them. This phase could also be composed of several steps depending on how the model was built, it may need performing the testing in several phases or considering the IPC structure in some specific manner. Once the model is tested, its results are evaluated. How the evaluation is conducted largely depends on the final objectives of the user, as we will see later.

In the next subsection we present the overview of the methods found in the literature to perform the ACHP in the IPC. As mentioned above, the creation of a classification model implies the use of several components, phases or steps. In order to normalize and structure the presentation of the methods used to build classification models to tackle the AHCP in the IPC we use the following components:

- **Classification method**
- **Features**
- **Hierarchy**
- **Evaluation**

We explain each component in more detail in the next sections, and then in section 4.7 we present the schematized overview of works in the literature for the AHCP in the IPC.

4.3 Classification Method

The field of text classification (TC) has been greatly developed during the past decades, because of that a variety of algorithms has been created. We present and describe here in a general way the main classification methods used in the literature for tackling the AHCP in the IPC. The formal and deep mathematical details of each of them can be found in the literature of machine learning and data mining [5][29][33][43][51][74].

Naïve Bayes. The naïve Bayes (NB) classifier is a simple probabilistic classifier based on applying Bayes' theorem with strong ("naive") independence assumptions. In simple terms, the NB classifier assumes that the presence (or absence) of a particular feature in a category is unrelated to the presence (or absence) of any other feature [37]. When training the classifier, the probabilities of each feature belonging to every category are estimated. When testing the classifier, the previously estimated probabilities are used to determine the probabilities that a document belongs to various categories. There are in essence two ways of estimating such probabilities [42]: the multi-variate Bernoulli model (where the features are considered in a document only as present or not present), and the multinomial model (where the features considered are the number of times they appear). The NB is easy to implement and despite its independence assumptions, it performs generally well in TC tasks.

***k*-Nearest Neighbors.** The k-nearest neighbors (kNN) classifier is a type of instance-based method. It encapsulates all the training data in order to use them later in the test phase. When a test document is to be classified, the kNN looks in the stored training data for the k most similar documents (neighbors) to it. Commonly, similarity is computed using a distance metric based on the feature distributions of the documents. The suggested category of the test document can then be estimated from the neighboring documents by weighting their contributions according to their distance [77]. Even if the kNN classifier relies on the whole training data to perform classification, it can be trained to find the optimal number of neighbors k as well as the best similarity metric. This method is very popular in TC tasks, where it performs generally well. There are many versions of this algorithm, depending on how the similarities and weights are computed.

Support Vector Machines. A support vector machine (SVM) [11] performs classification by constructing a hyperplane that optimally separates the training documents into two categories. The hyperplane is defined over the feature space of the documents, where they are represented as vectors. During training the classifier identifies the hyperplane with longest margin that separates the training documents into two categories. During testing, the classifier uses that hyperplane to decide which category a new document belongs to. SVMs are powerful algorithms to perform TC. They can handle a large number of features without loosing generality, and can easily be extended to the multi-label classification scenario.

Artificial Neural Networks. An artificial neural network (ANN) [30] consists of a network of many simple processing units interconnected between them with varying connection weights. The units are usually positioned in successive layers. Used for classification, a network layer receives an input in the form of features representing a document, processes it and gives an output to the next layer, and so on, until the final layer outputs the category(ies) of the document. During training, the method assigns and updates the weights to each unit by using the categorized trained data trying to minimize the categorization error. During testing, the network processes the features of the test document across the units and layers and outputs the categories. There exists a large number of versions of this method.

A particular version of ANN is the Universal Feature Extractor (UFEX) [60] algorithm. This method is a kind of one-layer ANN, which receives as an input a vector of features representing a document, and then outputs a set of categories for it. The training phase is done by a greedy update of the weights in each unit of the network, where each unit represents a category expressed as a vector of features (or category descriptor). When a document from the training set is assigned incorrectly to a category, the algorithm updates both category descriptors: the one of the true category (to force a correct classification) and the one of the wrong category (to avoid that similar documents reach that category).

Another version of ANN is the Winnow [39] algorithm. Winnow is a perceptron-like algorithm that uses a multiplicative scheme for updating the weights in the network units. This method could be extended to a multi-label scenario by learning a set of several hyperplanes at the same time.

Decision Trees. Decision tree (DT) algorithms [49] classify a document by following a set of classification rules. The rules indicate when a feature, a set of features or the absence of a feature are good indicators that a document belongs to a certain category. During training the algorithm learns such rules from the training data, where the rules are ordered in a tree-like structure, from more general to more specific rules. During testing the algorithms apply the rules to conduct the classification.

Logistic Regression. The logistic regression (LR) model performs classification by determining the impact of multiple independent variables (features) presented simultaneously to predict one of two categories (binary classification, similarly than with SVM). The probabilities describing the possible category are modeled as a function of the features using a logistic function. During training, logistic regression forms a best fitting equation or function using the maximum likelihood method, which maximizes the probability of classifying the training documents into the appropriate category by updating a set of regression coefficients. During testing, a test document, expressed as a vector of features, is multiplied by the regression coefficients and the model outputs the probability of the document belonging to one of the two categories. This method is very powerful for TC tasks, it can handle a large number of features without loosing generality, and can easily be extended to the multi-label classification scenario.

Minimizer of the Reconstruction Error. The Minimizer of the Reconstruction Error (mRE) [26][27] performs classification using the reconstruction errors provided by a set of projection matrices. In the training phase, it first builds a term-document matrix per category. Then, it performs a principal component analysis for each category matrix and obtain a projection matrix per category. During testing, a new test document is first projected using the reconstruction matrices, then it is reconstructed used the same matrices and the error between the reconstructed document and the original one is measured. The projection matrix that minimizes the error of reconstruction assigns the category. This model could be directly extended to a multi-label scenario by using thresholds to define the confidence of assigning a category to a document.

There are other classifiers that could be used inside a AHCP system. We do not intend to mention all the alternatives here, rather we mention only the most common, well-known or studied methods. When a different classification method is used in a specific system we will mention it and refer to the corresponding work for further details.

4.4 Features

There are many kinds of possible features to extract from the textual content of a patent. Among the most commonly used for TC tasks are: words, context words, word n-grams, phrases, character n-grams, and links. Except for the character n-grams, words are the basic block of construction (they are built of words). Words could be simply defined as sequences of characters (strings) separated by blanks. Context words for a given word w, are the words that co-occur in a patent together with w. Word n-grams are ordered sequences of words. Phrases are sequences of words following a syntactic scheme. Character n-grams are ordered sequences of characters. Links are words or sequences of words that make a reference to other patents or documents. The previous features are used to build a representation of the patent except for the links, which are used to extract information from related patents.

Patents, as we have seen in section 3.1, are structured and divided into a number of sections: the bibliographical data, the title, the abstract, the claims and the description. Then, the above described features (except for the links that could be extracted only from the bibliographical data) could be extracted from one, a portion of one, several or all the sections.

Once the features are extracted from the textual content, there are several preprocessing steps that could be conducted, as explained in the first part of this section: stop word removal (SWR), stemming, lemmatization, feature selection and vocabulary construction. The first three options are language dependant, and there exist several ways of performing these tasks. Stop word removal could be done by comparing a word with a list of already known stop words in a given language. Stemming [48] and lemmatization are related tasks; they try to reduce inflected (or sometimes derived) words to their root form in a given language. Lemmatization is more complex since it involves subtasks such as understanding the context and determining the part of speech for a word. Feature selection is usually independent of the language, and there is a collection of methods such as [78][23]: document frequency (DF), information gain (IG), mutual information gain, χ^2, etc.

After preprocessing, the resulting features are used to represent the patent in a format that the classification method can understand. That is done usually by expressing the patent as a vector of feature weights (named *vector space model* or VSM) that reflects the importance of each feature regarding the patent. There are several weighting schemes, the most common are: binary, term frequency (TF), term frequency inverse document frequency (TF-IDF), entropy and BM25 [41]. In the binary weighting each feature is expressed only as 1 or 0, if it is present or not in the patent. In the TF weighting each feature is counted the number of times it appears in the patent. In the TF-IDF weighting, the TF weighting is multiplied by the inverse of the number of times the feature appears in the whole patent collection (IDF). Entropy is based on information theory ideas and is a most sophisticated weighting scheme. Entropy gives higher weight for features that appear fewer times in a small number of patents, while it gives lower weight for features that appear many times along the collection of patents. BM25 indeed refers to a family of weighting schemas using different components and parameters. It is usually estimated using a logarithmic version of the IDF multiplied by the frequency of the feature which is normalized by the length of the patent and the average length of patents along the collection.

With the document representation done, there is still a last step of feature extraction, where several of the original features are combined to create a new set of reduced combined features. There is a collection of methods to perform this [43]: latent semantic indexing (LSI) [13], principal component analysis (PCA)

[5], linear discriminant analysis (LiDA) [61], non-negative matrix factorization (NMF) [53], latent Dirichlet allocation (LDA) [6], etc.

During training there are also a number of possibilities when considering several categories of each patent in the training data (the multi-label issue). Following the definition by Tsoumakas et al. [65] there are two ways to do it: problem transformation (PT), and algorithm adaptation (AA).

The methods following the PT approach are algorithm independent. They transform the multi-label task into one or more single-label classification tasks. As an example consider the following set of patents with their corresponding sets of categories: $\{(\mathbf{d}_1, \{c_1, c_2\}), (\mathbf{d}_2, \{c_1\}), (\mathbf{d}_3, \{c_1, c_2, c_3\})\}$. One way to transform this set into a single-label set is by copying each patent in each one of the categories it has assigned, this would produce a new set as follows: $\{(\mathbf{d}_{1a}, \{c_1\}), (\mathbf{d}_{1b}, \{c_2\}), (\mathbf{d}_2, \{c_1\}), (\mathbf{d}_{3a}, \{c_1\}), (\mathbf{d}_{3b}, \{c_2\}), (\mathbf{d}_{3c}, \{c_3\})\}$. A second possibility is to select at random only one category for the patents with more than one category assigned, this would produce a new set of patents as follows: $\{(\mathbf{d}_1, \{c_2\}), (\mathbf{d}_2, \{c_1\}), (\mathbf{d}_3, \{c_1\})\}$. Another alternative is to simple ignore the examples with multiple categories, as follows: $\{(\mathbf{d}_2, \{c_1\})\}$.

The methods following the AA approach extend specific learning algorithms in order to handle multi-label data directly. These methods usually learn at once the complete set of labels for all the patents. Following this approach, several well known methods have been adapted to handle multi-label data, such as SVM [17], decision trees [10] and k-NN [80].

4.5 Hierarchy

The AHCP task in section 4.1 was defined to classify patents over the hierarchy structure Υ, in our case the IPC. In general there are two approaches to use the structure when building the classification model: flat and hierarchical. The flat approach ignores completely the IPC. It simply trains a classification model in the desired level of the IPC and the predictions always concern that level.

The hierarchical approach could indeed be implemented in several ways using the IPC structure. Following the definitions by Silla and Freitas [55], the possibilities are: local classifier per node (LCN), local classifier per level (LCL), local classifier per parent node (LCPN) and global classifier (GC). In the LCN, a base binary classification method is trained for each category (node) of the IPC, and it decides if a test patent belongs or not to that category (and the classification is conducted only on the children nodes of the category assigned). In the LCL, a multi-class classification method is trained in each level of the IPC, and it decides to which categories in a given level a test patent belongs to (restricting the classification to the children nodes of the categories assigned in the previous level). In the LCPN, a multi-class classifier is trained in each node that is not a leaf, and it decides to which of its children categories belongs a test patent. In the GC, a single classifier considering all the IPC structure at once is created, and it predicts all the possible categories for a test patent at once.

In both cases, flat and hierarchical, the output could be single-label or multi-label, i.e. only assigning one category to the patent or several. As we have seen in section 3, the AHCP task is by nature multi-label. However, some systems restrict their output only to the most probable category to simplify the task.

Using the previous alternatives to include the hierarchy, the training and testing of the model could be also done in a single phase (SP) or in multiple phases (MP). In the single-phase approach, both the training and test phases are done only by using the training or test data only once, respectively. In the multi-phase approach, during the training phase the training patents are read several times to refine the classification model [3]. During the test phase, the predictions for each test patent are also refined based on ranking methods or combinations of several outputs [76].

Finally, it is important to determine the level of classification in the IPC for an AHCP system. The different levels impose different complexities, the lower the level the more difficult the task is. The levels are specified in section 2.

4.6 Evaluation

The output of an AHCP system is the category matrix $\mathbf{V} = \nu_1, \nu_2, \ldots, \nu_k$. That is, the collection of assigned categories for the patent test set. Once the system has provided all the categories for the test set, these results are then evaluated to measure the performance of the system. There are several performance measures, among the most used are: accuracy (Acc), precision (P), recall (R), F1-measure, mean average precision (MAP) and Hamming loss (H-loss). Accuracy is the percentage of correctly classified documents. There is a version of this measure called parent accuracy (PAcc). The PAcc is the Acc measured for each category node that has children in a hierarchy, and then the Acc is assigned to the corresponding children of such categories. Precision is the number of correctly classified positive documents divided by the number of documents classified by the system as positive. Recall is the number of correctly classified positive documents divided by the number of positive documents in the test data. In this case, the positive class is considered as the specific category that is being evaluated and the negative class includes all the other categories. F1-measure is the harmonic mean of precision and recall. P, R and F1-measure can be computed per individual patent and then averaged, i.e. micro-averaged (Mi-P, Mi-R, Mi-F1); or per complete category and then averaged, i.e. macro-averaged (Ma-P, Ma-R, Ma-F1). They could also be computed depending on the order of the categories returned by a system. These measures are defined as P@N, R@N and F1@N, where N indicates the number of sorted categories (from 1 to N) to consider when computing the measure. Finally, they could be also computed in a hierarchical way (hP, hR, hF1), to consider the classification in the different levels of a hierarchy, and in that way discount wrong assignments to categories lower in the hierarchy. MAP is the mean of the average precision over the test set, understood as the correct categories for a patent ranked by order. H-loss is the mean of the percentages of the wrong assigned categories to the total number of true categories for each patent in the test set. This loss could also be computed in a hierarchical way (Δ-loss), considering the loss along the hierarchy. We refer to Silla and Freitas [55], Sokolova and Lapalme [57], and Tsoumakas et al. [65] for a review of these measures applied in multi-class, multi-label and hierarchical scenarios.

The previous measures take into account the output of the AHCP system to compare with the true categories of the test patent. In this sense they measure the efficacy or correctness of the system. However, it is also expected that any AHCP system performs its task efficiently, i.e. it does not take a very long time to execute the training phase and/or the testing phase. This is usually done by estimating the computational complexity of the methods involved in the two phases (how many single operations the system needs to do its job), or by estimating the real time the system takes to perform the training and testing phases under a specific computer architecture.

Any evaluation measure should be checked for statistical significance, in order to ensure that a given performance is not produced by chance. There are several statistical tests, such as: t-test, Friedman test, McNemar test, Wilcoxon signed-ranks test, etc. We refer to the work of Demšar [14] for the use of statistical tests in classification tasks.

To conduct training, testing and evaluation, a collection of patents is needed. There are some datasets used to evaluate an AHCP system, such as: the WIPO-alpha dataset, the WIPO-de dataset and the CLEF-IP 2010 and 2011 datasets.

The WIPO-alpha collection [72] consists of patent applications submitted to WIPO under the Patent Cooperation Treaty (PCT). Each of these patents includes a title, a set of bibliographical data (except references), an abstract, a claims section, and a long description. The patents are in XML format (as seen in section 3.1), in English, and were published between 1998 and 2002. The collection is composed of 75,250 patents (46,324 for training and 28,926 for testing). These patents are distributed over 5,000 categories in the top four IPC levels: 8 sections, 114 classes, 451 subclasses, and 4,427 main groups.

The documents in the WIPO-de collection [72] were extracted from the DEPAROM source and were published between 1987 and 2002. The patents are written in German and also presented in XML format with the same structure as the ones in the WIPO-alpha dataset. The collection is composed of 117,246 patents. The collection is divided in training and test sets differently for the two top levels of the IPC hierarchy. At the class level there are 50,555 patents for training and 21,271 for testing. At the subclass level there are 84,822 patents for training and 26,006 for testing. These patents are distributed over 120 classes and 598 subclasses of the IPC.

The CLEF-IP 2010 [47] collection consists of patents in XML format in three languages: English, German and French. Each patent in this collection includes a title, a set of bibliographical data, an abstract, a claims section and a long description. These patents are mostly patents submitted to EPO. The collection is divided in about 1.3 millions of patents for training (with the proportions of 68% in English, 24% in German and 8% in French), and 2,000 patents for testing (1,468 in English, 409 in German and 123 in French). The patents are distributed across the complete IPC.

The CLEF-IP 2011 [46] collection is based on the CLEF-IP 2010 dataset. This dataset contains the patents of the CLEF-IP 2010 collection and 200,000 additional patents submitted to WIPO in its training set. The patents in this collection have the same XML format and structure as the ones in the CLEF-IP 2010 dataset, and there are about the same proportions of patents for English, German and French. The test set is composed of 3,000 patents (1,000 in each language). The patents are distributed across the complete IPC.

One last thing to consider when evaluating an AHCP system is the language it could process: mono-lingual (MoL), multi-lingual (MuL) or cross-lingual (CoL).

4.7 Comparison Between Different Systems for the AHCP in the IPC

Table 5 summarizes the components described in the previous sections and some of the alternatives for each one of them.

Table 5. Summary of the several components that could be used in the AHCP in the IPC. For explanation of the acronyms we refer to the corresponding section. In case a component is not completely defined in this chapter, we refer to the corresponding work for further details.

Component	Alternatives
Classification Method (CM)	NB, kNN, SVM, ANN, UFEX, Winnow, DT, LR, mRE, others
Features	**Features:** Words, context words, words n-grams, phrases, links, others **Sections of patents:** Title, abstract, description, claims, bibliographical data **Preprocessing:** SWR, stemming, lemmatization, other **Feature selection:** DF, IG, χ^2, others **Feature weighting:** Binary, TF, TF-IDF, entropy, BM25, others **Feature extraction:** LSI, PCA, LiDA, NMF, LDA, others **Multi-label consideration:** PT, AA
Hierarchy	**Hierarchy use:** Flat, hierarchical (LCN, LCL, LCPN or GC) **Output:** SL, ML **Level of classification in IPC:** class, subclass, main group, subgroup **Phases of classification:** SP, MP
Evaluation	**Dataset:** WIPO-alpha, WIPO-de, CLEF-IP 2010, CLEF-IP 2011, others **Language capability:** MoL, MuL, CoL **Evaluation measure:** Acc, PAcc, (Mi-, Ma- or h)P, (Mi-, Ma- or h)R, (Mi-, Ma- or h)F1-measure, MAP, H-loss, Δ-loss, others **Efficiency:** Complexity, computing time **Statistical test:** t-test, Friedman test, Wilcoxon signed-ranks test, others

Using the components summarized in table 5, in tables 6, 7, 8 and 9 we present a schematized summary of the several works found in the literature for the AHCP in the IPC.

In addition to the works described in the tables below, there are a set of overview papers regarding the AHCP in the IPC. Firstly there are two overview papers related with the classification tasks in the CLEF-IP 2010 and CLEF-IP 2011 workshops. These tasks used the corresponding datasets mentioned in section 4.6. The goal of each task was to classify the corresponding test sets, which consist of patents written in three languages: English, German and French (see section 4.6 for details). The overviews of the tasks are presented in [47] for CLEF-IP 2010 and in [46] for CLEF-IP 2011.

For the CLEF-IP 2010 classification task, the goal was to classify the test patents up to the subclass level of the IPC. There were seven participants submitting a total of 27 runs. The runs were variations of their corresponding systems (using different internal parameters). The organizers evaluated the performance of the submitted runs using the following measures: P@1, P@5, P@10, P@25, P@50, R@5, R@25, R@50, F1@5, F1@25, F1@50 and MAP. The results of the evaluation are presented per language (English, German and French) and as an average over the three languages. The organizers of this task sorted the performances using the P@5, R@5 and F1@5 measures.

Table 6. Overview of existing literature for the AHCP in the IPC. We try to detail as much as possible each component. If one of them is not listed for a given work is because it is not used, mentioned or considered in the corresponding work.

Work	Details
Aiolli et al. [1]	**Classification Method**: GPLM (generalized preference learning model) **Features:** Words **Sections of patents:** Title, abstract and first 300 words of description (all combined) **Preprocessing:** SWR and Porter stemming **Feature weighting:** Cosine normalized TF-IDF **Hierarchy use:** LCN **Output:** ML (variable) **Level of classification in IPC:** Subclass **Phases of classification:** SP **Dataset:** WIPO-alpha **Language capability:** MoL (English) **Evaluation measure:** 3-Layered Mi-F1. Best performance 0.5298 **Efficiency:** Linear on training **Statistical test:** Standard deviation
Beney [2]	**Classification Method**: Balanced Winnow **Features:** Words or linguistic triplets **Sections of patents:** Title or abstract or names or description (each section separated) **Output:** ML **Level of classification in IPC:** Class and Subclass **Phases of classification:** SP **Dataset:** CLEF-IP 2010 **Language capability:** MuL (English, German, French) **Evaluation measure:** Mi-F1. Best performance (using words+triplets in combination with title+abstract+names) 0.77 at the class level and (using words+title+abstract+names) 0.68 at the subclass level **Efficiency:** about 9 hours for training **Statistical test:** Standard deviation

Table 6. *Continued*

Work	Details
Cai & Hofmann [7]	**Classification Method**: hSVM (hierarchical SVM) **Features:** Words **Sections of patents:** Title and claims (combined) **Feature weighting:** Term normalization **Hierarchy use:** GC **Output:** SL (only the main category) **Level of classification in IPC:** Main group **Phases of classification:** SP **Dataset:** WIPO-alpha using 3-fold cross validation over the whole dataset **Language capability:** MoL (English) **Evaluation measure:** Acc, P, taxonomy-based loss (Δ-loss), parent accuracy (PAcc) Best performance Acc=0.38, P=0.49, Δ-loss=1.23, PAcc=0.65 **Efficiency:** 2,200 seconds for training
Chen & Chang [9]	**Classification Method**: SVM and kNN **Features:** Words **Sections of patents:** Title and claims (combined) **Preprocessing:** SWR and Porter stemming **Feature selection:** Inverse category frequency (TF-ICF) to select 1,040 features **Feature weighting:** TF-IDF **Hierarchy use:** LCL **Output:** ML in the first two phases (11 and 37 respectively), SL in the final decision (only the main category) **Level of classification in IPC:** Subgroup **Phases of classification:** MP. Three phases for training and testing Two initial phases with SVM and one final with kNN **Dataset:** A subset of WIPO-alpha (21,104 patents, 12,042 for training and 9,062 for testing) **Language capability:** MoL (English) **Evaluation measure:** Acc Top (main category). Best performance 0.36
Derieux et al. [15]	**Classification Method**: SVM **Features:** Words and phrases **Sections of patents:** Title, abstract, description and claims (all combined) **Preprocessing:** SWR, Part-Of-Spech tagging, lemmatization and polysemy filtering **Feature weighting:** Based on the section of the patent **Hierarchy use:** Flat **Output:** ML (20 categories) **Level of classification in IPC:** Subclass **Phases of classification:** MP. Two phases for training and testing **Dataset:** CLEF-IP 2010. Subset of training set (670,000 patents in English, 240,000 patents in German and 75,000 in French). The complete test set. **Language capability:** MuL (English, German, French) **Evaluation measure:** P@N. Best performance P@5=0.97 in English, P@5=0.96 in German and P@5=0.94 in French

Table 7. Continuation of table 6

Work	Details
Fall et al. [20]	**Classification Method**: SVM or NB or kNN or SNoW **Features:** Words **Sections of patents:** (a) Title or (b) claims (separate) (c) 300 first words of titles, inventors, applicants, abstracts and descriptions (combined) (d) titles, inventors, applicants, and abstracts (combined) **Preprocessing:** SWR and stemming **Feature selection:** IG **Feature weighting:** Binary **Multi-label consideration:** PT. Each patent is considered in each category where it is assigned, or it is considered in its main category. **Hierarchy use:** Flat **Output:** ML (3 categories) **Level of classification in IPC:** Class and subclass **Phases of classification:** SP **Dataset:** WIPO-alpha **Language capability:** MoL (English) **Evaluation measure:** Acc Top, Acc Three and Acc All Best performance at class level, Acc Top=0.55 (SVM, set of features (c)), Acc Three=0.79 (NB, 300 words), Acc All=0.63(NB, set of features (c)) Best performance at subclass level, Acc top=0.41 (SVM, set of features (c)), Acc Three=0.62 (kNN, 300 words), Acc All=0.48(SVM, set of features (c))
Fall et al. [21]	**Classification Method**: NB or kNN or SVM or LLSF (Linear Least Squares Fit) **Features:** Words **Sections of patents:** Two sets (a) the first 300 different words of the titles, inventors, applicants and claims sections. (b) the first 300 different words of the titles, inventors, companies and descriptions **Preprocessing:** SWR and stemming **Feature weighting:** Binary (kNN) and TF (NB and SVM) **Hierarchy use:** Flat **Output:** ML **Level of classification in IPC:** Class and cubclass **Phases of classification:** SP **Dataset:** WIPO-de **Language capability:** MoL (German) **Evaluation measure:** Acc Top, Acc Three and Acc All. Best performance Acc Top=0.65 (LLSF, set (b) of features) at class level Acc Three=0.86 (LLSF, set (b) of features) at class level Acc All=0.76 (LLSF, set (b) of features) at class level Acc Top=0.56 (LLSF, set (b) of features) at subclass level Acc Three=0.78 (LLSF, set (b) of features) at subclass level Acc All=0.71 (LLSF, set (b) of features) at subclass level
Gomez & Moens [27]	**Classification Method**: mRE (Minimizer of the Reconstruction Error) **Features:** Words **Sections of patents:** Title, abstract and 30 first lines of description (all combined) **Preprocessing:** SWR **Feature weighting:** Normalized TF-IDF **Multi-label consideration:** PT. Each patent is considered in each category where it is assigned **Hierarchy use:** Flat **Output:** SL (only the main category) **Level of classification in IPC:** Section **Phases of classification:** SP **Dataset:** WIPO-alpha, WIPO-de **Language capability:** MuL (English, German) **Evaluation measure:** Acc, Ma-F1. Best performance Acc=0.74, Ma-F1=0.72 for WIPO-alpha Best performance Acc=0.69, Ma-F1=0.68 for WIPO-de **Efficiency:** Quasi-linear on training
Guyot et al. [28]	**Classification Method**: Winnow **Features:** Words and context words (collocations) **Sections of patents:** Inventor, applicant, title, abstract, claims, first 4,000 characters of description (all combined) **Preprocessing:** SWR **Feature selection:** TF (remove words that appear less than 4 times), and keep collocations that appear more than 16 times **Hierarchy use:** Flat **Output:** ML **Level of classification in IPC:** Subclass **Phases of classification:** SP **Dataset:** CLEF-IP 2010 **Language capability:** MuL (English, German, French) **Evaluation measure:** MAP and P@N Best performance MAP=0.79, P@1=.83 (average over the three languages) **Efficiency:** About 3 hours for training and 3 minutes for testing (common PC)

Table 8. Continuation of table 6

Work	Details
Hofmann & Cai [31]	**Classification Method**: SVM **Features:** Words **Sections of patents:** Title and claims (combined) **Feature weighting:** Normalization **Hierarchy use:** GC **Output:** SL (only the main category) **Level of classification in IPC:** Main group **Phases of classification:** SP **Dataset:** Section D of WIPO-alpha (1,710 patents) using 3-fold cross validation **Language capability:** MoL (English) **Evaluation measure:** Acc, Δ-loss. Best performance Acc=0.30, Δ-loss=1.21
Rousu et al. [50]	**Classification Method**: H-M^3 (Maximum Margin Hierarchical Multilabel Classifier) **Features:** Words **Feature weighting:** TF-IDF **Multi-label consideration:** AA **Hierarchy use:** GC **Output:** ML **Level of classification in IPC:** Main group **Phases of classification:** SP **Dataset:** Section D of WIPO-alpha (1,372 patents for training and 358 for testing) **Language capability:** MoL (English) **Evaluation measure:** Mi-F1, Δ-loss. Best performance Mi-F1 = 0.76, Δ-loss=1.67 **Efficiency:** Linear
Seeger 2006 [52]	**Classification Method**: Kernel classification model **Features:** Words **Sections of patents:** Title and claims (combined) **Preprocessing:** SWR and Porter stemming **Feature weighting:** Normalization **Multi-label consideration:** AA **Hierarchy use:** GC **Output:** ML **Level of classification in IPC:** Main group **Phases of classification:** SP **Dataset:** WIPO-alpha (experiments per section A to H) with 3 different splits **Language capability:** MoL (English) **Evaluation measure:** Acc, P, taxo-loss Best performance Acc=0.37, P=0.49, taxo-loss=1.25 **Efficiency:** Linear for training
Teodoro et al. [59]	**Classification Method**: kNN **Features:** Words **Sections of patents:** Title, abstract, claims and links (combined) **Feature weighting:** Normalized BM25 **Hierarchy use:** Flat **Output:** ML **Level of classification in IPC:** Subgroup **Phases of classification:** MP. No training phase. Two phases for testing **Dataset:** PAJ (2,382,595 patents in Japanese) and USPTO (889,116 patents in English) for training. 633 abstracts in English and 639 in Japanese for testing **Language capability:** MoL (English), CoL (Classify papers written in Japanese, using patents written in English) **Evaluation measure:** MAP. Best performance 0.68 at subclass level, 0.5 at main group level and 0.3 at subgroup level

Table 9. Continuation of table 6

Work	Details
Tikk et al. [60]	**Classification Method**: UFEX **Features:** Words or phrases **Sections of patents:** Title, inventor, applicant, abstract, claims (combined) **Feature selection:** DF (disregard words appearing in less 2 patents and in more than 25% of the training set) **Feature weighting:** Entropy **Multi-label consideration:** AA **Hierarchy use:** LCN **Output:** ML (3 categories) **Level of classification in IPC:** Class, subclass and main group **Phases of classification:** SP **Dataset:** WIPO-alpha, WIPO-de **Language capability:** MuL (English, German) **Evaluation measure:** Acc Top, Acc Three and Acc All. Best performance Acc Top=0.66, Acc Three=0.89, Acc All=0.76 for WIPO-alpha at class level Acc Top=0.55, Acc Three=0.79, Acc All=0.66 for WIPO-alpha at subclass level Acc Top=0.38, Acc Three=0.60, Acc All=0.51 for WIPO-alpha at main group level Acc Top=0.65, Acc Three=0.87, Acc All=0.75 for WIPO-de at class level Acc Top=0.55, Acc Three=0.78, Acc All=0.67 for WIPO-de at subclass level Acc Top=0.38, Acc Three=0.57, Acc All=0.51 for WIPO-de at main group level **Efficiency:** 2 hours 40 minutes for training on a PC (2Ghz, 1GB in RAM)
Trappey et al. [62]	**Classification Method**: NN **Features:** Phrases (made of correlated words) **Preprocessing:** SWR **Feature selection:** DF (the 67 most frequent words are selected) **Hierarchy use:** Flat **Output:** SL (only the main category) **Level of classification in IPC:** Main group and subgroup **Phases of classification:** SP **Dataset:** Class B25 from WIPO-alpha (124 patents for testing) **Language capability:** MoL (English) **Evaluation measure:** P Best performance 0.92 at main group level, 0.9 at subgroup level
Verbene et al. [68]	**Classification Method**: Winnow **Features:** Words and dependency triplets (two words and their dependency) **Sections of patents:** Abstract **Feature weighting:** Binary **Multi-label consideration:** AA **Hierarchy use:** Flat **Output:** ML **Level of classification in IPC:** Subclass **Phases of classification:** SP **Dataset:** CLEF-IP 2010. Only the English part for training and the whole test set **Language capability:** CoL (Classify patents written in English, German or French, using patents written in English) **Evaluation measure:** P, R, F1, MAP Best performance (using words+triplets) P=0.62, R=0.52, F1=0.56, MAP=0.69 (average over the three languages) **Efficiency:** 2 hours for training
Verbene et al.[67]	**Classification Method**: Winnow **Features:** Words, dependency triplets, links **Sections of patents:** Abstract, metadata, description and first 400 words of description (combined) **Feature weighting:** Binary **Multi-label consideration:** MP. Two phases for testing (voting scheme using categories from linked patents) **Hierarchy use:** Flat **Output:** ML **Level of classification in IPC:** Subclass **Phases of classification:** SP **Dataset:** CLEF-IP 2011 **Language capability:** MoL (English) **Evaluation measure:** P, R, F1 Best performance (words+abstract+description) P=0.74 (words+triplets+abstract+400 words of description) R=0.86 (words+abstract+description) F1=0.71

For the CLEF-IP 2010 classification task [47], the participant group from Simple Shift (described as Guyot et al. [28] in the tables above) obtained the best performance. However, as a matter of fact, the general performance of the systems for this task varies depending on which measure to consider. The other published works related with this task and described in the tables are the ones of Beney [2], Derieux et al. [15] and Verberne et al. [68].

In the CLEF-IP 2011 [46], there were two classification tasks: the first was to classify the test patents in the subclass level of the IPC, the second was to classify the test patents in the subgroup level of the IPC provided the real subclass of each patent (i.e. to refine the classification). There were only two participants with a total 25 runs for both tasks. The organizers evaluated the performance of the submitted runs using the following measures: P@1, P@5, R@1, R@5, F1@1 and F1@5. For the subclass level the best results were from the group of the Information Foraging Lab of the Radboud Universiteit Nijmegen (described as Verberne et al. [67] in the tables above). For the subgroup level the best results reported in the overview paper were from the group WISEnut Inc with P@5≈0.32 for English, P@5≈0.29 for German and P@5≈0.27 for French. However, we were unable to access the published work of this group.

There exist also two overview papers regarding the classification task in the NTCIR-7 [44] and NTCIR-8 [45] workshops. The task was the same in both workshops: to classify research papers (not patents) using the IPC, but the AHCP systems had to be trained using patents. In NTCIR-7 the classification was done in the subgroup level, while in NTCIR-8 the classification was done in the subclass, main group and subgroup levels. The task was multi-lingual and cross-lingual, using patents and papers written in Japanese and English. There were four subtasks: classification of research papers written in English using a system trained with patents written in English; classification of research papers written in Japanese using a system trained with patents written in Japanese; classification of research papers written in Japanese using a system trained with patents written in English (J2E subtask); and classification of research papers written in English using a system trained with patents written in Japanese (E2J). The organizers provided the participants with a dataset for training of about 8 million patents. 7 millions of those patents were written in Japanese and from there 3.5 million of patents were automatically translated, the remaining 1 million of patents were written in English. For testing they provided 644 research papers in English and Japanese. For the NTCIR-7 workshop there were twelve participants submitting a total of 50 runs for the first three subtasks (no submissions for the E2J subtask). The best performances were obtained for the Japanese subtask with a MAP=0.44, for the English subtask with a MAP=0.49, and for the J2E subtask with a MAP=0.44.

In the case of the NTCIR-8 workshop there were six participants submitting a total of 101 runs for the first three subtasks (no submissions for the E2J subtask). The best performances at the subclass level were obtained for the Japanese subtask with a MAP=0.8, for the English subtask with a MAP=0.72, and for the J2E subtask with a MAP=0.71; at the main group level for the Japanese subtask a MAP=0.64, for the English subtask a MAP=0.55, and for the J2E subtask a MAP=0.5 were cited; and at the subgroup level for the Japanese subtask a MAP=0.45, for the English subtask a MAP=0.37, and for the J2E subtask a MAP=0.30 were obtained.

We could observe that the CLEF-IP and NTCIR classification tasks have a predominant natural language processing (NLP) background and follow an information retrieval (IR) approach for the AHCP in the IPC. The IR approach sees the problem as retrieving the most relevant categories for a given test patent, rather than classifying the patent in a set of categories.

From all the tables above and the description of the overview papers, we can observe the diversity of methodologies used to perform the AHCP in the IPC. One interesting point to highlight is that most of the authors agree that the use of more data for training is always beneficial to improve the performance of any AHCP system. They also agree that the deeper the level of classification in the IPC structure, the more complex the problem is and the worse the results are. As a matter of fact it is noticeable that there is still not a clear solution to the general problem of AHCP in the IPC. The descriptions of works show a large variety of results using different classification methods, features, sections of the patents, datasets, levels of classification and evaluation measures. Each group of authors claims to obtain better results based on their proposed framework. It is easily observable that there are still several aspects of the AHCP in the IPC that present a lack of agreement between researchers. What classifier method, features, preprocessing and section(s) of the patents are the best for the classification task and what is the best way of using the IPC structure are still open questions that are not completely nor clearly answered by any methodology. The results largely vary depending on the components used to implement a system and the evaluation measures used to estimate its performance. In this direction, there is a lack of a standard framework to evaluate the AHCP systems. We observe from the presented works in the above tables that most of the researchers use ad-hoc datasets and evaluation measures. There are few exceptions: the evaluation under the CLEF-IP 2010 and CLEF-IP 2011 tasks, which used the corresponding CLEF-IP datasets and used the same evaluation standard; and the works by Fall et al. (2003) [20], Fall et al. (2004) [21], Tikk et al. [60] and Chen&Chang [9], where the authors use the complete WIPO-alpha and WIPO-de datasets as they were originally defined, and use the same evaluation measures. In those cases it is possible to compare systems. Besides these, the comparison is rather complicated. We conclude that a standard framework of evaluation is required. In addition, deeper studies and experiments regarding the alternatives of the aforementioned components of an AHCP system are necessary, in order to better understand the effects of each one of them in the performance of the systems. Moreover, a better description of the complexity or computing times of the methods employed in a given AHCP system is desirable. This task is a large-scale task, and scalability of the methods should be considered, since the system would need to deal with thousands of patents per day.

5 Conclusions and Perspectives for the AHCP in the IPC

In this chapter we have surveyed and presented a revision of several works found in the literature for the automated hierarchical classification of patents (AHCP) in the International Patent Classification (IPC) hierarchy. This task, as we have seen throughout the sections of the chapter, is a very hard problem. It involves issues regarding the complex structure of the IPC, concerning its imbalanced distribution of categories, and its dynamical nature, together with particularities from the patents as written documents, from distributions of words to issues with the language used.

We have presented as well a series of components that can be included in an AHCP system. We then used these components to describe the works presented in the literature that deal with the task. We could observe from those works that there are still holes and lacks in the definition, scope and evaluation of the task. The works in the literature vary largely in their methodologies but also in their results, where the absence of a standard of evaluation (both in data and measures) is noticeable. It is also common that the works do not present the details used for the implementation of their methods, such as complexity, which would help to understand the scalability and usability of the algorithms.

This is one of the main concerns here. The definition of a standard framework adopted generally to evaluate AHCP systems. This framerwork should include standard datasets and evaluation measures, defined under the agreement of users and designer of the systems and considering both efficacy and efficiency.

Furthermore, most of the works devoted to the AHCP in the IPC are based on classical and traditional methods and use straightforward methodologies. There are several alternatives for the components described in section 4 that are not yet (well) explored for the ACHP in the IPC. Some authors claim in their works that SVMs are slow to train, but efficient implementations of the linear version of this classifier already exist [8][22][54]. There also exist other methods that consider the complex dependencies in a hierarchy and the multi-label nature of some problems which could be applied here [16][58][79][80]. The refinement of the final prediction of the categories to be assigned to a patent or the inclusion of several phases during training is also not well studied [3]. However, our guess is that given the large-scale nature of the AHCP in the IPC, some methods that impose dependencies or refinement during training or testing could have issues with efficiency. In that sense, more research is expected to fully exploit all the knowledge at hand when dealing with a complex hierarchy such as the IPC.

Additionally, the effects of the alternatives for feature selection and feature extraction are not yet clearly understood for the AHCP in the IPC. Some works apply basic statistics for feature selection, like DF or TF, but the use and scope of these methods in the task are still unclear. Feature extraction is even less explored, we have not found the application of methods like LiDA, NMF or LDA. In both cases of feature selection and extraction, it would be interesting to investigate how to use those methods along the hierarchy [25] in order to find features, topics or components describing the categories (and possibly the relations among them).

Acknowledgments. This research was supported partially by the KU Leuven project RADICAL (GOA 12/003).

References

1. Aiolli, F., Cardin, R., Sebastiani, F., Sperduti, A.: Preferential text classification: Learning algorithms and evaluation measures. Information Retrieval 12(5), 559–580 (2009)
2. Beney, J.: LCI-INSA linguistic experiment for CLEF-IP classification track. In: CLEF (Notebook Papers/LABs/Workshops) (2010)
3. Bennett, P.N., Nguyen, N.: 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 (2009)
4. Benzineb, K., Guyot, J.: Automated patent classification. In: Lupu, M., Mayer, K., Tait, J., Trippe, A.J. (eds.) Current Challenges in Patent Information Retrieval. The Information Retrieval Series, vol. 29, pp. 239–261. Springer (2011)
5. Bishop, C.M., Nasrabadi, N.M.: Pattern Recognition and Machine Learning. Springer (2006)
6. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet allocation. Journal of Machine Learning Research 3, 993–1022 (2003)
7. Cai, L., Hofmann, T.: Hierarchical document categorization with support vector machines. In: Proceedings of the 13th ACM International Conference on Information and Knowledge Management, pp. 78–87. ACM (2004)
8. Chang, C.C., Lin, C.J.: LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology 2(3), 27:1–27:27 (2011)
9. Chen, Y.L., Chang, Y.C.: A three-phase method for patent classification. Information Processing and Management 48(6), 1017–1030 (2012)
10. Clare, A.J., King, R.D.: Knowledge discovery in multi-label phenotype data. In: Siebes, A., De Raedt, L. (eds.) PKDD 2001. LNCS (LNAI), vol. 2168, pp. 42–53. Springer, Heidelberg (2001)
11. Cortes, C., Vapnik, V.: Support-vector networks. Machine Learning 20(3), 273–297 (1995)
12. CPC: Website of the Cooperative Patent Classification, `http://www.cooperativepatentclassification.org/index.html` (2013) (accessed: January 01, 2014)
13. Deerwester, S.C., Dumais, S.T., Landauer, T.K., Furnas, G.W., Harshman, R.A.: Indexing by latent semantic analysis. Journal of the American Society for Information Science 41(6), 391–407 (1990)
14. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research 7, 1–30 (2006)
15. Derieux, F., Bobeica, M., Pois, D., Raysz, J.P.: Combining semantics and statistics for patent classification. In: CLEF (Notebook Papers/LABs/Workshops) (2010)
16. Deschacht, K., Moens, M.F.: Efficient hierarchical entity classifier using conditional random fields. In: Proceedings of the 2nd Workshop on Ontology Learning and Population, pp. 33–40 (2006)
17. Elisseeff, A., Weston, J.: A kernel method for multi-labelled classification. In: Dietterich, T.G., Becker, S., Ghahramani, Z. (eds.) Advances in Neural Information Processing Systems, vol. 14, pp. 681–687. MIT (2002)

18. EPO: Website of the European Patent Office, http://www.epo.org/ (accessed: January 1, 2014)
19. Fall, C.J., Benzineb, K.: Literature survey: Issues to be considered in the automatic classification of patents. Tech. rep., World Intellectual Property Organization (October 2002)
20. Fall, C.J., Törcsvári, A., Benzineb, K., Karetka, G.: Automated categorization in the international patent classification. SIGIR Forum 37(1), 10–25 (2003)
21. Fall, C., Törcsvári, A., Fiévet, P., Karetka, G.: Automated categorization of German-language patent documents. Expert Systems with Applications 26(2), 269–277 (2004)
22. Fan, R.E., Chang, K.W., Hsieh, C.J., Wang, X.R., Lin, C.J.: LIBLINEAR: A library for large linear classification. Journal of Machine Learning Research 9, 1871–1874 (2008)
23. Forman, G.: An extensive empirical study of feature selection metrics for text classification. Journal of Machine Learning Research 3, 1289–1305 (2003)
24. Gomez, J.C., Boiy, E., Moens, M.F.: Highly discriminative statistical features for email classification. Knowledge and Information Systems 31(1), 23–53 (2012)
25. Gomez, J.C., Moens, M.-F.: Hierarchical classification of web documents by stratified discriminant analysis. In: Salampasis, M., Larsen, B. (eds.) IRFC 2012. LNCS, vol. 7356, pp. 94–108. Springer, Heidelberg (2012)
26. Gomez, J.C., Moens, M.F.: PCA document reconstruction for email classification. Computational Statistics & Data Analysis 56(3), 741–751 (2012)
27. Gomez, J.C., Moens, M.F.: Minimizer of the reconstruction error for multi-class document categorization. Expert Systems with Applications 41(3), 861–868 (2014)
28. Guyot, J., Benzineb, K., Falquet, G., Shift, S.: myclass: A mature tool for patent classification. In: CLEF (Notebook Papers/LABs/Workshops) (2010)
29. Han, J., Kamber, M., Pei, J.: Data Mining: Concepts and Techniques. Morgan Kaufmann (2006)
30. Haykin, S.: Neural Networks: A Comprehensive Foundation. Prentice Hall (1994)
31. Hofmann, T., Cai, L., Ciaramita, M.: Learning with taxonomies: Classifying documents and words. In: NIPS Workshop on Syntax, Semantics, and Statistics (2003)
32. Hull, D.A.: Stemming algorithms: A case study for detailed evaluation. Journal of the American Society for Information Science 47(1), 70–84 (1996)
33. Kantardzic, M.: Data Mining: Concepts, Models, Methods, and Algorithms. John Wiley & Sons (2011)
34. Seutter, C.H.A.K.M., Beney, J.G.: Multi-classification of patent applications with Winnow. In: Broy, M., Zamulin, A.V. (eds.) PSI 2003. LNCS, vol. 2890, pp. 546–555. Springer, Heidelberg (2004)
35. Krier, M., Zaccà, F.: Automatic categorisation applications at the European patent office. World Patent Information 24(3), 187–196 (2002)
36. Larkey, L.S.: A patent search and classification system. In: Proceedings of the 4th ACM Conference on Digital Libraries, pp. 179–187. ACM (1999)
37. Lewis, D.D.: Naive (Bayes) at forty: The independence assumption in information retrieval. In: Nédellec, C., Rouveirol, C. (eds.) ECML 1998. LNCS (LNAI), vol. 1398, pp. 4–15. Springer, Heidelberg (1998)
38. Li, W.: Random texts exhibit Zipf's-law-like word frequency distribution. IEEE Transactions on Information Theory 38(6), 1842–1845 (1992)
39. Littlestone, N.: Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. Machine Learning 2(4), 285–318 (1988)
40. Lupu, M., Hanbury, A.: Patent retrieval. Foundations and Trends in Information Retrieval 7(1), 1–97 (2013)

41. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press (2008)
42. McCallum, A., Nigam, K.: A comparison of event models for naive Bayes text classification. In: AAAI 1998 Workshop on Learning for Text Categorization, vol. 752, pp. 41–48. AAAI Press (1998)
43. Murphy, K.P.: Machine Learning: A Probabilistic Perspective. The MIT Press (2012)
44. Nanba, H., Fujii, A., Iwayama, M., Hashimoto, T.: Overview of the patent mining task at the NTCIR-7 workshop. In: Proceedings of the NII Test Collection for IR Systems-7. NTCIR (2008)
45. Nanba, H., Fujii, A., Iwayama, M., Hashimoto, T.: Overview of the patent mining task at the NTCIR-8 workshop. In: Proceedings of the NII Test Collection for IR Systems-8. NTCIR (2010)
46. Piroi, F., Lupu, M., Hanbury, A., Zenz, V.: CLEF-IP 2011: Retrieval in the intellectual property domain. In: Petras, V., Forner, P., Clough, P.D. (eds.) Proceedings of CLEF 2011 (Notebook Papers/Labs/Workshop) (2011)
47. Piroi, F.: CLEF-IP 2010: Classification task evaluation summary. Tech. Rep. IRF-TR-2010-00005, Information Retrieval Facility (August 2010)
48. Porter, M.F.: An algorithm for suffix stripping. Program: Electronic Library and Information Systems 14(3), 130–137 (1980)
49. Quinlan, J.R.: Induction of decision trees. Machine Learning 1(1), 81–106 (1986)
50. Rousu, J., Saunders, C., Szedmak, S., Shawe-Taylor, J.: Kernel-based learning of hierarchical multilabel classification models. Journal of Machine Learning Research 7, 1601–1626 (2006)
51. Sebastiani, F.: Machine learning in automated text categorization. ACM Computing Surveys 34(1), 1–47 (2002)
52. Seeger, M.: Cross-validation optimization for large scale hierarchical classification kernel methods. In: Advances in Neural Information Processing Systems, pp. 1233–1240 (2006)
53. Seung, D., Lee, L.: Algorithms for non-negative matrix factorization. Advances in Neural Information Processing Systems 13, 556–562 (2001)
54. Shalev-Shwartz, S., Singer, Y., Srebro, N.: Pegasos: Primal estimated sub-gradient solver for svm. In: Proceedings of the 24th International Conference on Machine Learning, pp. 807–814. ACM (2007)
55. Silla Jr., C.N., Freitas, A.A.: A survey of hierarchical classification across different application domains. Data Mining and Knowledge Discovery 22(1-2), 31–72 (2011)
56. Smith, H.: Automation of patent classification. World Patent Information 24(4), 269–271 (2002)
57. Sokolova, M., Lapalme, G.: A systematic analysis of performance measures for classification tasks. Information Processing and Management 45(4), 427–437 (2009)
58. Tang, L., Rajan, S., Narayanan, V.K.: Large scale multi-label classification via metalabeler. In: Proceedings of the 18th International Conference on World Wide Web, pp. 211–220. ACM (2009)
59. Teodoro, D., Gobeill, J., Pasche, E., Ruch, P., Vishnyakova, D., Lovis, C.: Automatic IPC encoding and novelty tracking for effective patent mining. In: Proceedings of the 8th NTCIR Workshop Meeting, pp. 309–317. National Institute of Informatics Japan (2010)
60. Tikk, D., Biró, G., Yang, J.: Experiment with a hierarchical text categorization method on WIPO patent collections. In: Attoh-Okine, N., Ayyub, B. (eds.) Applied Research in Uncertainty Modeling and Analysis. International Series in Intelligent Technologies, vol. 20, pp. 283–302. Springer (2005)

61. Torkkola, K.: Linear discriminant analysis in document classification. In: IEEE ICDM Workshop on Text Mining, pp. 800–806. IEEE (2001)
62. Trappey, A.J.C., Hsu, F.C., Trappey, C.V., Lin, C.I.: Development of a patent document classification and search platform using a back-propagation network. Expert Systems with Applications 31(4), 755–765 (2006)
63. Tseng, Y.H., Lin, C.J., Lin, Y.I.: Text mining techniques for patent analysis. Information Processing and Management 43(5), 1216–1247 (2007)
64. Tsochantaridis, I., Joachims, T., Hofmann, T., Altun, Y.: Large margin methods for structured and interdependent output variables. Journal of Machine Learning Research 6, 1453–1484 (2005)
65. Tsoumakas, G., Katakis, I., Vlahavas, I.: Mining multi-label data. In: Data Mining and Knowledge Discovery Handbook, pp. 667–685. Springer (2010)
66. USPTO: Website of the United States Patent and Trademark Office (2014), `http://www.uspto.gov/` (accessed January 01, 2014)
67. Verberne, S., D'hondt, E.: Patent classification experiments with the Linguistic Classification System LCS in CLEF-IP 2011. In: Proceedings of CLEF 2011 (Notebook Papers/Labs/Workshop) (2011)
68. Verberne, S., Vogel, M., D'hondt, E.: Patent classification experiments with the linguistic classification system LCS. In: CLEF (Notebook Papers/LABs/Workshops) (2010)
69. Vishwanathan, S.V., Schraudolph, N.N., Smola, A.J.: Step size adaptation in reproducing kernel hilbert space. Journal of Machine Learning Research 7, 1107–1133 (2006)
70. Wanner, L., Baeza-Yates, R., Brügmann, S., Codina, J., Diallo, B., Escorsa, E., Giereth, M., Kompatsiaris, Y., Papadopoulos, S., Pianta, E., Piella, G., Puhlmann, I., Rao, G., Rotard, M., Schoester, P., Serafini, L., Zervaki, V.: Towards content-oriented patent document processing. World Patent Information 30(1), 21–33 (2008)
71. Webster, J.J., Kit, C.: Tokenization as the initial phase in NLP. In: Proceedings of the 14th Conference on Computational Linguistics, pp. 1106–1110. ACL (1992)
72. WIPO: WIPO-alpha readme (2009), `http://www.wipo.int/classifications/ipc/en/ITsupport/Categorization/dataset/wipo-alpha-readme.html` (accessed: January 01, 2014)
73. WIPO: Website of the World Intellectual Property Organization (2014), `http://www.wipo.int/export/sites/www/classifications/ipc/en/guide/guide_ipc.pdf` (accessed: January 01, 2014)
74. Witten, I.H., Frank, E., Hall, M.A.: Data Mining: Practical Machine Learning Tools and Techniques. Elsevier (2011)
75. Wu, F., Zhang, J., Honavar, V.: Learning classifiers using hierarchically structured class taxonomies. In: Zucker, J.-D., Saitta, L. (eds.) SARA 2005. LNCS (LNAI), vol. 3607, pp. 313–320. Springer, Heidelberg (2005)
76. Xiao, T., Cao, F., Li, T., Song, G., Zhou, K., Zhu, J., Wang, H.: kNN and reranking models for English patent mining at NTICR-7. In: Proceedings of the 7th NTCIR Workshop Meeting. National Institute of Informatics Japan (2008)
77. Yang, Y.: An evaluation of statistical approaches to text categorization. Information Retrieval 1(1-2), 69–90 (1999)
78. Yang, Y., Pedersen, J.O.: A comparative study on feature selection in text categorization. In: Proceedings of the 14th International Conference on Machine Learning, pp. 412–420. Morgan Kaufmann (1997)

79. Zhang, M.L., Zhou, Z.H.: Multilabel neural networks with applications to functional genomics and text categorization. IEEE Transactions on Knowledge and Data Engineering 18(10), 1338–1351 (2006)
80. Zhang, M.L., Zhou, Z.H.: ML-kNN: A lazy learning approach to multi-label learning. Pattern Recognition 40(7), 2038–2048 (2007)