Chapter 11
Patent Classification on Subgroup Level Using Balanced Winnow

Eva D'hondt, Suzan Verberne, Nelleke Oostdijk, and Lou Boves

Abstract In the past decade research into automated patent classification has mainly focused on the higher levels of International Patent Classification (IPC) hierarchy. The patent community has expressed a need for more precise classification to better aid current pre-classification and retrieval efforts (Benzineb and Guyot, Current challenges in patent information retrieval. Springer, New York, pp 239–261, 2011). In this chapter we investigate the three main difficulties associated with automated classification on the lowest level in the IPC, i.e. subgroup level. In an effort to improve classification accuracy on this level, we (1) compare flat classification with a two-step hierarchical system which models the IPC hierarchy and (2) examine the impact of combining unigrams with PoS-filtered skipgrams on both the subclass and subgroup levels. We present experiments on English patent abstracts from the well-known WIPO-alpha benchmark data set, as well as from the more realistic CLEF-IP 2010 data set. We find that the flat and hierarchical classification approaches achieve similar performance on a small data set but that the latter is much more feasible under real-life conditions. Additionally, we find that combining unigram and skipgram features leads to similar and highly significant improvements in classification performance (over unigram-only features) on both the subclass and subgroup levels, but only if sufficient training data is available.

11.1 Introduction

In the last decades, patents have gained an enormous economic importance. Patent filing rates increase every year, and patent attorneys and examiners of the various patent offices are straining to deal with the large number of applications submitted every day. In this situation, automating (part of) the process by which

E. D'hondt (✉)
Radboud University Nijmegen, Nijmegen, The Netherlands

Laboratoire d'Informatique pour la Mécanique et les Sciences de l'Ingénieur, Orsay, France
e-mail: eva.dhondt@limsi.fr

S. Verberne • N. Oostdijk • L. Boves
Radboud University Nijmegen, Nijmegen, The Netherlands
e-mail: s.verberne@cs.ru.nl; n.oostdijk@let.ru.nl; l.boves@let.ru.nl

M. Lupu et al. (eds.), *Current Challenges in Patent Information Retrieval*,
The Information Retrieval Series 37, DOI 10.1007/978-3-662-53817-3_11

incoming applications are processed has great economic value [17]. Automatic patent classification, that is, automatically assigning relevant category labels from the International Patent Classification (IPC) taxonomy (see below) to an incoming document, may be an invaluable asset in both the *pre-classification* and *examination* phases of the patent granting process.

During the pre-classification stage, a patent application is examined by a person who has a general knowledge about all technological fields and—most importantly—has expert knowledge of the patent classification system. This expert then routes the application to the department(s) that specialises in the technical fields relevant to the invention described in the application [28]. At the European Patent Office (EPO), there have been attempts to automate this process [17], but due to low accuracy scores, pre-classification is currently limited to the higher (more abstract) levels of the IPC taxonomy.

In the examination phase, a patent examiner will perform a high-precision, interactive search to find documents that describe inventions similar to the one described in the application, in a bid to determine the existence of prior art for this invention. Prior art queries usually consist of field-specific terminology with specialised (low-level) IPC labels as query terms. In this phase, a fine-grained, consistent and high-quality patent classification is indispensable [27]. The research presented in this chapter aims to implement, improve and evaluate automated classification on lower (more specific) levels in the taxonomy, thus allowing for more specific suggestions during the pre-classification and examination processes.

In most patent offices, incoming patents are categorised and indexed using the International Patent Classification (IPC) system, a complex hierarchical category structure which covers all areas of technology. The IPC is a manually constructed taxonomy, which has been updated and refined over the last 30 years and is used in the patent offices of over 90 countries. It currently comprises five levels, of increasingly fine granularity: sections, classes, subclasses, groups and subgroups. The latest instantiation of the IPC (IPC-2015.01) comprises eight sections, about 130 classes, about 640 subclasses, around 7400 main groups and approximately 64,000 subgroups.

Most of the previous research on automatic patent classification has focused on classification at the higher levels in the IPC hierarchy, i.e. class and subclass levels. State-of-the-art classification results are around 62 % F1@5[1] on the subclass level [22, 23]. With about 130 and about 640 different categories, respectively, classification at the class and subclass levels is challenging, but computationally feasible for most classification algorithms.

The more detailed group and subgroup levels are generally deemed extremely difficult to classify properly for three reasons:

First, the categories on the lower levels generally show a large amount of overlap [31], and only part of the information in the document is potentially useful in distinguishing a category from related categories. Let us illustrate this with an

[1] 'F1@5' denotes the F1 score evaluated at rank 5.

example: subclass *A47C* comprises *chairs, sofas, beds*, and subclass *A47J* holds *kitchen equipment*. On subgroup level, the differences between categories are more subtle; they correspond to a small difference in the implementation or use of the invention, e.g. subgroup *A47C 17/12* covers *sofas changeable to beds by tilting or extending the armrests*, while subgroup *A47C 17/14* holds *sofas changeable to beds by removing parts only*. Consequently, the overlap of textual features between categories is likely to be much larger on the lowest level than on higher levels in the hierarchy.

The issue of overlapping categories is further complicated by the peculiar language use in patents. To increase the scope of legal protection, patent attorneys use obfuscating language to describe the inventions, so that a mundane object like a *pump* becomes a *fluid transportation device*. The abundance of vague terms in the patent corpora makes it extra hard to distinguish between categories that already have a high overlap. In previous research, D'hondt et al. [7] found that adding more precise (phrasal) features such as skipgrams[2] to unigram (word) features improves classification at the IPC *class* level. It is not known if skipgrams would also capture the supposedly more subtle differences on the lower levels in the hierarchy.

Second, the large number of categories on lower levels in the IPC results in a computationally expensive classification task with severe scalability issues [1]. A common approach to deal with a large number of categories in a multi-level taxonomy, which are characterised by fine-grained distinctions, is a *hierarchical classification* method (as opposed to *flat classification*) [9]. Hierarchical classifiers can consist of one integrated classifier that is trained with knowledge of the structure of a taxonomy [2] or a set of classifiers that predict category labels in individual nodes of a (predefined) taxonomy [26]. Integrated and distributed hierarchical classifiers can be implemented in many different ways. In this chapter we will use the most common architecture of a distributed classifier: the 'local classifier per parent node approach' proposed by Silla and Freitas [26]. In this architecture each parent node in the category hierarchy corresponds with a multi-class classifier, which is trained to distinguish between the child nodes. The training material for a classifier is selected through the 'siblings' policy: when training a classifier to distinguish one daughter, e.g. subclass 'A01B' from all other daughters (subclasses) in the same 'world', i.e. class 'A01', all examples of 'A01B' are selected as positive training material, while the examples with labels 'A01L', 'A01D', ...serve as negative training material.

In the test phase, it is common to use a top-down class-prediction approach: when a document is classified by a hierarchical system, the output of the classifier at the parent nodes influences the classification conducted at the child nodes at the next level of the hierarchy. The classification process can be accelerated substantially if

[2] 'Skipgrams' are sequences of N words in a text, in which up to M intervening words may be deleted. Thus, a 2-skip-2-gram is a sequence of two words (bigram) that are no more than two words apart in a text. For example, from the example sentence *'I like to drive.'*, the following set of 2-skip-2-grams can be generated: `I_like, I_to, I_drive, like_to, like_drive, to_drive`.

the procedure at the next lower level is limited to the daughters of the categories that had the highest probability of being correct at the higher levels. When applied to the group and subgroup levels in patent classification, where on average a group comprises 12 subgroups, reducing the classifiers on lower levels to the most promising mother nodes simplifies the classification procedure substantially, when compared to the 64,000 subgroups that a flat classifier must distinguish.

Another advantage of a hierarchical classifier may be that, given the different training sets and the differences in overlap between categories, classifiers on lower levels might be able to select different and more focused features than classifiers that operate on a higher level of a taxonomy. Consider a system that needs to distinguish between 'clothes' and 'gardening tools' on a higher level and—within the 'clothes' category—between 'bikinis' and 'swimming trunks' on the lower level. Terms such as 'water', 'cover' and 'texture' will be informative features for the high-level classifier, but less so for the low-level classifier. We would expect the latter classifier to select more features that focus on the (smaller) differences between the categories, such as 'man' versus 'woman', 'top', etc.

A drawback of top-down hierarchical classifier systems is that they are susceptible to the *propagation of error* problem [18]: an erroneous hard decision at an upper level will propagate down the hierarchy, making it impossible to arrive at the correct low-level category label. Several solutions have been proposed to counter the error propagation, of which the most common is to backtrack when the classification scores on lower levels become too low. However, as is well known from syntactic parsing, backtracking mechanisms quickly become unwieldy. As a consequence it is claimed that single-level (flat) methods are more efficient than hierarchical methods, but that hierarchical methods are generally more accurate [4].

The *third* reason classification on group and subgroup levels is generally deemed too difficult is that the relative sparseness in the number of documents per category creates training difficulties [11]. Most data sets available for research in text classification have a certain degree of skewness of their distribution. In the patent domain, where technological categories move with different evolutionary speed—which entails shifts in the number of applications per category over time [8]—we found that a small proportion of the categories comprise the bulk of the documents [7]. The impact of the skewness of the distribution of documents over categories on a specific classification task is difficult to predict and may depend on the type of classifier that is being used.

We hypothesise that the scalability issues mentioned by Benzineb and Guyot [1] and the large degree of overlap between subgroups mentioned by Widodo [31] can both be addressed by using hierarchical, rather than flat, classifiers. In this chapter we examine the impact of flat and hierarchical approaches on the classification of abstracts of patent applications on the deepest (subgroup) level in the IPC hierarchy. In addition, we investigate the impact of different text representations (unigrams versus skipgrams) on the classification performance. By performing experiments on two data sets of different sizes, we will also address the issues caused by the skewness of the distributions in data sets that are available for scientific research.

In concrete terms, this chapter attempts to answer two fundamental questions:

1. How do flat and hierarchical classification methods compare in classifying on the subgroup level with the WIPO-alpha set? For both methods we use the Balanced Winnow classification algorithm. Following Chen and Chang [4], we simplify the five-level hierarchical classification problem in the IPC hierarchy to a two-level problem: subclass and subgroup. To avoid the problem of the propagation of error, we do not make a selection of top-n categories on the subclass level, but we will consider all possible branches in the classification tree. As proposed by Dumais and Chen [9], we convert classification scores to posterior probabilities for class membership. The posteriors from the subclass and subgroup levels are then combined to obtain class membership probabilities at the subgroup level.
2. Can we improve the classification on subgroup level by adding phrasal features to unigram features? Since previous research [7] indicated that phrasal features are only effective given a large amount of training data, we conducted this analysis not only on the (relatively small) WIPO-alpha corpus but also on the larger CLEF-IP 2010 corpus.

By virtue of the fact that we perform experiments on two data sets of different sizes, we will be able to shed light on the interaction between, and the relative importance of, the three problems with patent classification mentioned in the literature: too large a number of categories, sparseness of documents per category and high similarity between categories.

11.2 Related Work

For a detailed overview of the literature concerning the impact of different text representations on patent classification, we refer the reader to [7]. Here, we will focus on the use of flat or hierarchical classifiers.

An extensive overview of the various methods used for hierarchical classification in multiple application domains can be found in [26]. In this section we will limit ourselves to approaches to text classification in the patent domain.

As mentioned in the introduction, methods for hierarchical text classification fall into two subgroups: (1) methods that consist of one integrated classifier that uses the (hierarchical) relations between the categories as additional information next to textual content and (2) a multi-level approach with different sets of classifiers on different levels in a taxonomy. In Sects. 11.2.1 and 11.2.2, we discuss literature about applying both types of methods to classification in the patent domain. In Sect. 11.2.3 we describe an approach for combining classification scores in hierarchical classification, which has not yet been used in the patent domain before.

11.2.1 Training One Classifier with Information from the Hierarchy

Cai and Hofmann [2] propose a hierarchical classification method based on support vector machines (SVM). Their method does not perform classification in two or more steps, but encodes the hierarchical information in the description of categories and then performs flat classification. Cai and Hofmann [2] do this by extending the multi-class SVM algorithm with the possibility of representing each category with an attribute vector instead of a single category label. They encode the hierarchical relationships between the categories as attributes for the categories. They compare their hierarchical implementation of SVM to standard (flat) SVM in classification on the main group level for the WIPO-alpha collection. They find that their hierarchical approach gives similar accuracy to the standard SVM approach, but with the hierarchical approach the incorrectly assigned categories are closer to the correct categories in the taxonomy than with the standard approach.

Wang et al. [30] combine a top-down hierarchical classifier (as will be presented in Sect. 11.2.2) with a meta-classifier to arrive at more balanced rankings on the lowest level in a hierarchy. The meta-classifier takes *meta-samples* as features. These samples are feature vectors that encode information on the ‘path’ through the hierarchy to arrive at a low-level category, rather than the textual content of that category. They collect such information as the scores of the related base classifiers, the number of nodes on a path, the average scores of nodes along a path, etc., in a sparse vector. Wang et al. [30] evaluate their method on the [18] data set and find that it achieves a similar accuracy as flat classification systems.

11.2.2 Two-Step Classification

In the NTCIR-6 track, a special task was devoted to the two-level classification taxonomy used in the Japan Patent Office. The category set in the first level is an extension of IPC, in the form of a set of thematic categories. For example, the theme 2C088 is about ‘pinball game machines’ [18]. The categories on the second level denote the ‘viewpoint’ of the invention. Examples of viewpoints are purpose, means, function and effect. Each theme has a set of viewpoints and each viewpoint may consist of several elements, which are organised in a tree structure. For example, the theme 2C088 has a viewpoint AA ‘machine detail’, which has the element AA01 ‘vertical pinball machines’ [18]. The viewpoints with their elements are encoded as so-called F-terms in the patent. Li et al. [18] compared flat classification of F-terms using SVM to hierarchical classification using a variant of SVM called H-SVM [3]. They find that their method for hierarchical classification performed much worse than what they could achieve with flat classification. They suggest that the hierarchical relations among the classes are too complicated for the H-SVM algorithm.

Another branch of hierarchical classification systems explicitly exploits the hierarchical properties of the IPC taxonomy, either through user interaction or by combining classification output on different levels to predict labels on subgroup level.

The myClass classification tool [13] is a neural network implementation of the Balanced Winnow algorithm and achieved the highest accuracy in the CLEF-IP 2010 classification task (on subclass level) [22]. This tool uses a semi-automatic[3] method for classification on subgroup level [12]. A user is asked to select the correct labels from classification output on an intermediate level, such as subclass or main group. In a second step, the tool outputs subgroup labels within the selected (intermediate) categories.

Tikk et al. [28] propose a taxonomy-driven architecture for text classification called HITEC. They model the tree structure of the class hierarchy as a neural network. The categorisation of an incoming document is performed from the top of the hierarchy downwards. Going from top to bottom in the hierarchy, each level is followed by a so-called authorisation layer. The classifier determines the classification score of the document for all active category nodes at each level. Based on this score, the authorisation layer decides which categories on the next level are activated. In doing so, the authors use a novel *relaxed greedy algorithm*: Rather than activating only the category with the highest relevance score at each level, the system allows multiple categories to be active if their label scores are above a given threshold and within a given margin of variation from the highest label score. By thus widening the search, the authors expect to counter the propagation of error. However, the classification scores from the higher levels are not taken into account in calculating the classification scores for the lower levels. Consequently, the final rankings are based solely on the similarities between the test documents and the category models on the lowest levels, which might suffer from the fact that very few training documents are available for a large proportion of the categories. Tikk et al. [28] evaluate their method on the WIPO-alpha set. They classify documents on three levels: class, subclass and main group. They obtain excellent results with 53.25 % accuracy at the subclass level, which is 12 percentage points higher than the best-scoring setting reported in the reference paper by Fall and Benzineb [11]. On the main group level, Tikk et al. [28] achieve an accuracy of 36.89 %.

Chen and Chang [4] extend the work done by Tikk et al. [28] and were—to our knowledge—the first to classify on subgroup level. They develop a three-phase classification method which combines flat SVM classifiers at two different levels of the IPC hierarchy, namely, subclass and subgroup level, with a KNN classifier on the subgroup level. Their method takes four parameters $k1$–$k4$. In the first phase, a test document is classified on subclass level and a predetermined number of category labels are returned (variable $k1$). These subclass categories are then pooled together

[3] In its latest version myClass offers fully automatic classification on subgroup level [14]. However, as myClass is proprietary software, a detailed technical description of its current implementation has not been published.

to form a large 'world' in which a classifier is trained, this time on subgroup level. In the second phase, a predetermined number of category labels on subgroup level are returned (variable $k2$). The classifier that is needed for the first step can be built beforehand, but the classifier for the second step is variable and must be learned dynamically after the top-$k1$ subclasses have been identified. In the third phase of the algorithm by Chen and Chang [4], each subgroup from the top-$k2$ of subgroups is split in $k3$ clusters of documents using k-means clustering. Then, cosine similarity is calculated between the test document and the mean of each cluster. A KNN classifier with $k = k4$ is used to choose the most similar subgroup for the test document, i.e. the subgroup category with the most occurrences in the $k4$ most similar document clusters.

In a pretest phase, Chen and Chang [4] examine 'almost all combinations' (p. 11) of the parameters $k1$–$k4$ to determine the optimal combination with the highest accuracy. For this pretest, they use a subset of 400 documents from the test data. Their best-scoring setting ($k1 = 11$, $k2 = 37$, $k3 = 5$, $k4 = 169$) achieves a 36.07 % accuracy at the subgroup level.[4] Since they did not use a held-out development set for parameter tuning, these results can be considered an upper bound for classification performance with their three-phase method. For the sake of comparison, Chen and Chang [4] also re-implemented the HITEC classifier by Tikk et al. [28] and, using this system, they achieve 30.2 % for the same test set on subgroup level.

11.2.3 Combining the Classification Scores on Different Levels in the Hierarchy

As we saw in the previous two subsections, none of the approaches in previous work on hierarchical patent classification combines the scores of classifiers on different levels. The common approach is to let the output of the high-level classifier determine which classifiers on lower levels are activated [28], or what training material should be selected to train a classifier on the lower level [4]. In both cases, individual category scores do not have a direct impact on lower levels in the hierarchical classifier.

If we look outside the patent domain, however, we can find methods that combine classifier scores from different levels in a hierarchy. An example of this in a text classification task is [9], who performs Web page classification on a small two-level corpus of (summarised) Web pages, which consists of 13 categories on the first level and 150 categories on the second level. In order to be able to combine scores from different classifiers, they first derive posterior probabilities from SVM output scores. They then proceed to compare the impact of (1) thresholding on higher levels in the hierarchy (effectively minimising the number of categories to be examined

[4] They also report the accuracy of their algorithm without the third step ($k1 = 11, k2 = 1$): 20.2 %.

at the lower level) with (2) combining higher- and lower-level probabilities through multiplication and then thresholding on the final probabilities. Both methods achieve similar final rankings (of the top N results). Dumais and Chen [9] also compare the hierarchical systems with a flat (baseline) classification system. They find that hierarchical methods significantly outperform that baseline system.

11.3 Data Selection and Processing

11.3.1 Data Selection

In this section we describe the two patent corpora used for the experiments presented in Sects. 11.5 and 11.6. The WIPO-alpha data set is a well-known benchmark for patent classification, which was first made available by the World Intellectual Property Organization (WIPO) in 2002. Although it is a clean and often-used data set, it is fairly small compared to present standards. We therefore opted to run a second series of experiments on the CLEF-IP 2010 data set, which is more representative of a real-life patent corpus.

11.3.1.1 WIPO-Alpha Data Set

The English WIPO-alpha collection[5] consists of 75,250 patent documents (46,324 for training and 28,926 for testing) with their IPC category labels on subgroup level.[6] The documents were published between 1998 and 2002 and are labelled with the 7th version of the IPC.

From each patent document, we extracted the abstract section, using the information in the XML source. Since some subgroups have little to no training data, we used the same data selection criteria as [4][7]: we only selected subgroups that have a minimum of seven training documents. This selection step resulted in a corpus of 22,113 documents (12,883 for training and 9230 for testing). The corpus statistics after document selection in Table 11.1 show that there is a large variation in the number of documents (abstracts) in the different categories, both on subclass level and subgroup level. Moreover, 628 of the 1140 categories on subgroup level contain fewer than ten documents. Having only seven documents as positive examples for training a classifier is on—or below—the lower bound of what is needed to construct

[5] The collection can be downloaded at http://www.wipo.int/classifications/ipc/en/ITsupport/Categorization/dataset/index.html.

[6] Since IPC labels are hierarchical, i.e. contain information on parent nodes in the label, we can easily extract subclass labels from the subgroup labels.

[7] Unlike [4], we used the official training/test split as determined by the EPO. Our category selection was based on frequency counts over the training set only.

Table 11.1 Corpus statistics on the WIPO-alpha corpus after sample selection

	# of cat	av. # doc in cat (stdev)	av. # daughters (stdev)
Subclass	339	38.00 (53.19)	3.36[a] (4.36)
Subgroup	1140	11.30 (7.18)	n.a.

[a] 128 subclasses only have one subgroup daughter in the training set

a useful category model. But even with this lenient criterion, we were forced to discard more than 70 % of the documents in the WIPO-alpha collection.

All documents in the WIPO-alpha collection come with one primary (subgroup) category label (determining the field of application in which the invention is novel) and may have several secondary categories. In the following experiments, we only take the primary category labels into account, thus rendering the WIPO-alpha experiments into a mono-label, multi-class hierarchical classification problem.

11.3.1.2 CLEF-IP 2010 Data Set

The CLEF-IP 2010 data set[8] is a subset of the MAREC corpus[9] and was released as part of the CLEF-IP 2010 classification and prior art retrieval tracks. It features 2.6 million patent documents from the European Patent Office (EPO). These three million documents with content in English, German and French pertain to over one million patents,[10] from 1976 to 2002.

As with the WIPO-alpha corpus, we first extracted all abstracts from the patent documents and then applied data selection on the corpus. We used more stringent selection criteria than for the WIPO-alpha set: only subgroups with a minimum of 50 documents were included in the corpus subset. This cut-off was arbitrarily chosen to avoid data sparseness in the subgroup categories on the one hand, while on the other hand minimising the number of one-daughter subclass worlds. The resulting subset was then divided into training/test corpora with the same ratio as the WIPO-alpha split (60/40), with the additional criterion that all subgroups in the training set must contain at least 20 documents. This resulted in a corpus subset of 991,805 documents and a training and test set of 595,080 and 396,725 documents, respectively. Statistics on the CLEF-IP corpus after sample selection are given in Table 11.2. It shows that in the fairly large CLEF-IP data set, the distribution is very skewed. When making the train/test split, we tried to minimise the number of categories that might suffer from data sparseness. We therefore chose a split where only 493 of the 19,411 subgroup categories contain fewer than 30 training

[8] Available at http://www.ifs.tuwien.ac.at/~clef-ip/download/2010/index.shtml#data.

[9] Available at http://www.ifs.tuwien.ac.at/imp/marec.shtml.

[10] Unlike the WIPO-alpha data set, the CLEF-IP data set contains documents that refer to the same patent but in various stages of the granting process. Consequently, some of the extracted abstracts may be similar to each other.

Table 11.2 Corpus statistics on the CLEF-IP 2010 subset corpus after sample selection

	# of cat	av. # doc in cat (stdev)	av. # daughters (stdev)
Subclass	575	8028.4 (20,512.2)	33.8[a] (63.4)
Subgroup	19,441	237.5 (434.8)	n.a.

[a] 39 subclasses only have one subgroup daughter in the CLEF-IP 2010 subset

documents. For the CLEF-IP corpus, it also holds that fairly lenient data selection criteria in designing a classification experiment result in discarding almost 70 % of the documents. Please note the size difference between the two corpora: even after data selection there is—on average—six times more data available for a category on subgroup level in the CLEF-IP 2010 corpus than there is for a subclass category in the WIPO-alpha corpus.

Patent documents in the CLEF-IP 2010 data set may contain multiple labels and—unlike the WIPO-alpha set—have no information on primary versus secondary labels. We therefore included all labels, rendering the CLEF-IP experiments a multi-label, multi-category classification task. In consequence, the similarity between categories on both levels is likely to be higher since categories may share some training documents as positive examples during training.

11.3.2 Text Preprocessing and Feature Generation

While the WIPO-alpha corpus is a fairly clean text corpus and requires little preprocessing effort, the CLEF-IP 2010 corpus contains several data conversion errors which were solved using regular expressions.

After removing all XML markup from the extracted abstracts, we ran a Perl script to divide the running text into sentences, by splitting on end-of-sentence punctuation such as question marks and full stops. In order to minimise incorrect splitting of the technical texts that contain many acronyms and abbreviations, the Perl script was supplied with a list of common English abbreviations and a list containing abbreviations and acronyms that occur frequently in technical texts,[11] derived from the Specialist lexicon.[12]

The sentences in the WIPO-alpha and the CLEF-IP corpora were then further processed to generate lemmatised unigrams and skipgrams. In previous research [6, 7], we found that classification accuracy (on class level) is more improved by adding skipgrams which are filtered for specific parts of speech than by adding bigrams or dependency triples generated by a parser.

[11] Both the splitter and abbreviation file can be downloaded from https://sites.google.com/site/ekldhondt/downloads.

[12] The lexicon can be downloaded at http://lexsrv3.nlm.nih.gov/Specialist/Summary/lexicon.html.

To generate unigram and skipgram features, the preprocessed sentences were tagged using an in-house PoS tagger [29].[13] The tagger's statistical language models have been trained on the annotated subset of the British National Corpus. We opted for this particular tagger because it is highly customisable to new lexicons and word frequencies, which is essential when dealing with the patent domain: the language usage in patent documents can differ greatly from that in other genres. For example, the past participle *said* is often used to modify nouns as in *'for said claim'*. While this usage is very rare and archaic in general English, it is a very typical modifier in patent language. Consequently, a PoS tagger must be updated to account for these differences in language use. To this end we adapted the tagger with word frequency information and associated PoS tags from the AEGIR lexicon.[14] We did not retrain the N-gram language model of the tagger, since no PoS-tagged patent texts are available for that purpose. The words in the tagged output were also lemmatised using the AEGIR lexicon.

From the tagged output, we then generated two text representations using the filtering and lemmatisation procedure described in [6]: PoS-filtered words (only allowing nouns, verbs and adjectives) and PoS-filtered 2-skip-2-grams (only allowing combinations of nouns, verbs and/or adjectives). In the experiments described in this chapter, *unigrams* will refer to the PoS-filtered words only, while *unigrams* + *skipgrams* will refer to the combination of PoS-filtered words and PoS-filtered 2-skip-2-grams.

11.4 Classification Algorithms

In this section we first describe the training algorithm of the classifiers in both the flat and hierarchical classification approaches. Section 11.4.2 describes our approach to hierarchical classification on subgroup level in the IPC hierarchy.

11.4.1 Balanced Winnow Algorithm

We opted to use the Balanced Winnow classification algorithm implementation in the Linguistic Classification System (LCS), because it has been shown in previous work to be very fast and effective for large-scale text classification problems and to yield state-of-the-art results on text classification problems with many categories [6, 7, 16].

[13] Tokenisation was performed by the tagger.

[14] The AEGIR lexicon is part of the AEGIR parser, a hybrid dependency parser that is designed to parse technical texts, with a focus on patent text. For more information, see [20].

Preceding the actual training, there is a two-step term selection phase in which the most informative terms are selected for each category. In the first step (global term selection), selection is based on global frequency information, i.e. a term must appear in at least three documents in the training set and at least twice in those documents. In the second step (local term selection), we used the LTC algorithm [25] to calculate TF-IDF scores for the features per category. We then selected the top 1000 most informative features per category and aggregated them into the initial category models (a.k.a. class profiles).

(Balanced) Winnow is a mistake-driven learning algorithm, akin to the perceptron algorithm. The effect of learning during training is determined by four parameters: a promotion parameter α, a demotion parameter β and two threshold parameters θ^{+} and θ^{-}, which determine a threshold 'beam'.

In Balanced Winnow, each feature is given two weights (w^{+} and w^{-}), the sum of which is the Winnow weight. The terms are initialised with their winnow weights set to their TF-IDF scores. During training the weights w^{+} and w^{-} are only updated when a mistake occurs in classifying the training documents. The algorithm distinguishes two types of mistakes: (1) true label is not found and (2) wrong label is assigned. In the former case, the weights w^{+} of the active features are promoted by multiplying them with α, while the weights w^{-} of the active features are demoted by multiplying them with β (thus increasing the final Winnow weights of the active features). In case of error (2), the weights w^{+} of the active features are demoted by multiplying them with β, and the weights w^{-} are promoted by multiplication with α. The 'beam' determined by the θ parameters delineates an area where correct labelling is still considered a type 1 error, which leads to more weight updates.

In the test phase, when classifying a document d, the term vector representing d is checked against each category model, a.k.a. class profile, in the classifier and assigned a Winnow score for that category. This score is the sum of the Winnow scores for the individual terms in the term vector. In Sect. 11.4.3, we describe how we tuned the Winnow parameters.

11.4.2 Hierarchical Classifiers

11.4.2.1 System Architecture of the Hierarchical Classifiers

Following [4], our hierarchical approach to classification operates in a downward two-level hierarchy: On the first level, there is one classifier trained on a corpus-wide training set, annotated with IPC subclass information. In the case of the WIPO-alpha data set, this classifier distinguishes between 339 different (subclass) categories; for the CLEF-IP 2010 data set, it distinguishes between 575 different (subclass) categories. Hereafter we will refer to these classifiers as the *subclass classifiers*.

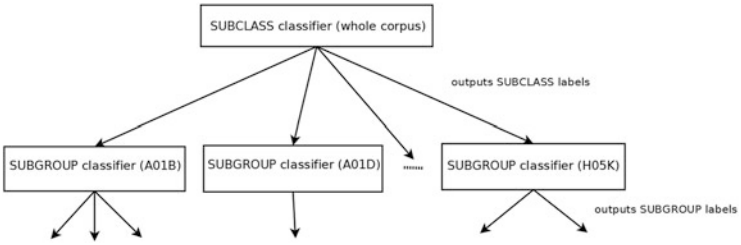
SUBCLASS classifier (whole corpus)
outputs SUBCLASS labels
SUBGROUP classifier (A01B)
SUBGROUP classifier (A01D)
....
SUBGROUP classifier (H05K)
outputs SUBGROUP labels

Fig. 11.1 Structure of a hierarchical classifier

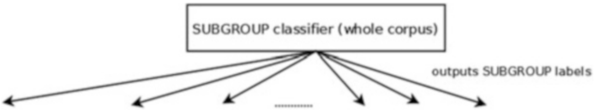

Fig. 11.2 Structure of a flat classifier

On the second level, for each subclass category a separate classifier is trained, which differentiates between the subgroup daughters in that subclass world.[15] A *subgroup classifier* is trained only on the training data available in a particular subclass world and yields classification scores for the different subgroup categories in that world. As was shown in Tables 11.1 and 11.2, the number of daughters in different subclass worlds can vary greatly. In our system the patent documents are always assigned a label on subgroup level; we do not assign labels on the intermediate group level. Figures 11.1 and 11.2 illustrate the architectures of a hierarchical and flat classifier, respectively. Each box in Fig. 11.1 refers to an individual flat classifier.

11.4.2.2 Normalisation and Converting Scores to Probabilities

During the test phase, a vector representing a test document is first scored by the subclass classifier and then by each of the subgroup classifiers. To arrive at a final ranking of subgroup labels, the scores of the classifiers on the two levels must be combined in a way that takes into account the differences in scoring ranges between the various classifiers.

We achieve this by transforming the Winnow scores of each document for each category into an estimate of the posterior probability that the document belongs in a given category. For that purpose, we used the sigmoid transformation proposed by Platt [24]. In the case of the subclass classifier for the WIPO-alpha set, each

[15] Please note that subclass worlds are the default context for training subgroup classifiers. In Sect. 11.5 we will also report additional experiments where subgroup categories were trained in larger contexts, i.e. class and section worlds.

document obtains 339 Winnow scores for as many subclasses, only one of which is correct. This leads to a substantial imbalance in the data for the logistic regression (for each relevant '1' score, there are 338 '0' scores) which we accounted for by the error weighting in [15] which we integrated in the implementation for finding the sigmoid proposed in [19].

Although the transformation of Winnow scores to probabilities by means of a continuously non-decreasing function cannot alter the rank order of the subclasses, it can increase or shrink the distance between the values assigned to subclasses. This becomes relevant when combining the probability scores derived from the subclass classifier with the probability scores from the different subgroup classifiers to achieve a final ranking on subgroup level.

To avoid a bias caused by the differences in score ranges between the subclass and the various subgroup classifiers—the subclass classifier scores generally span a wider range than those given by the subgroup classifiers—we decided to normalise the Winnow scores before transforming them to posterior probabilities. This was done using Batch Normalisation: for each classifier we calculated a linear function through which the Winnow scores for the training documents were mapped into the range $[0.0, 10.0]$. These linear functions were calculated by running a fivefold cross-validation over the training data available for that classifier and then mapping the complete set of scores into a range of 0 to 10 with the (original) maximum and minimum Winnow score in the complete set as anchor values.

A second bias that we wished to avoid is caused by the difference in the amount of training data on the two different levels: from Tables 11.1 and 11.2, it can be seen that the average number of documents available for training subclass classifiers is much larger than the number of documents for training subgroup classifiers. From this we can conjecture that the subclass classifier and the corresponding sigmoid function trained on subclass data are potentially better (in)formed than the individual classifiers and the corresponding sigmoids for the different subclass worlds. This hypothesis was confirmed by an analysis of the score distributions for categories on subgroup level. As mentioned above, Winnow scores from the subgroup classifiers are generally not widespread, and we found that—even after normalisation—the scores of relevant and irrelevant categories were quite similar. Consequently, the sigmoids fitted on this data may not yield accurate transformations from Winnow scores to posterior probabilities. We experimented with different definitions of the 'worlds' for training subgroup classifiers in which more training data was available, but we did not find significant improvements in the eventual classification performance.

We therefore decided to fine-tune the balance between the subclass and subgroup probability estimates to arrive at an optimal final ranking on subgroup level. We assigned weights by raising the subclass probabilities to power γ and the subgroup probabilities to power δ, respectively. For both data sets, we performed full-grid

Table 11.3 Winnow parameters for hierarchical and flat classifiers for the WIPO-alpha data set, determined after fivefold cross-validation tuning

	Subclass (hierar)	Subgroup (hierar)	Subgroup (flat)
α	1.06	1.03	1.06
β	0.91	0.98	0.91
θ^+	2.0	2.0	2.0
θ^-	1.0	1.0	1.0

Table 11.4 Winnow parameters for hierarchical and flat classifiers for the CLEF-IP 2010 data set, determined after fivefold cross-validation tuning

	Subclass (hierar)	Subgroup (hierar)	Subgroup (flat)
α	1.02	1.02	–
β	0.98	0.98	–
θ^+	2.0	2.0	–
θ^-	0.5	0.5	–

searches[16] on subsets from the cross-validation folds in the training procedure. Interestingly, similar patterns emerged for both the WIPO-alpha and the CLEF-IP data sets: To reach optimal ranking, the subclass probabilities should be raised to a relatively high power, while the subgroup probabilities should be raised to a very low power.[17] We arrived at the optimal balance by raising the subclass probabilities to the power of 1.5 (γ) and the subgroup probabilities to the power of 0.2 (δ).

11.4.3 Tuning

The classification parameters for the subclass classifiers, the subgroup classifiers and the flat classifiers were determined individually by tuning through fivefold cross-validation on a subset of the training data. All subgroup classifiers use the same parameters. These are the parameters that yielded the best overall results in an oracle experiment with fivefold cross-validation.[18] The resulting parameter settings are in Tables 11.3 and 11.4. With the exception of θ^-, the parameters for the subclass and subgroup classifiers in both corpora are very similar.

Note that we do not report any parameters for a flat subgroup classifier on the CLEF-IP 2010 data set: as mentioned in the introduction, the complexity of a 19,441-category (multi-label) classification problem causes severe scalability

[16] $\gamma \in \{0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0\}$, $\delta \in \{0.1,0.2,0.3,0.4,0.5\}$. For the WIPO-alpha hierarchical classifier, we optimised on success@rnk1. In the case of the (multi-label) CLEF-IP classifier, we optimised on the F1 accuracy score.

[17] By raising them to a high power, subclass probabilities 'shrink', i.e. result in lower probabilities which increases the distance between the high-scoring and intermediate labels. For the subgroup classifiers on the other hand, intermediate probabilities (from 0.6 onwards) are transformed into extremely high scores (between 0.9 and 1.0).

[18] In an oracle setting, documents are only tested against subgroup classifiers from the relevant subclass world(s).

issues [1]. Even on a server with two Intel© Xeon© E5-2660 Processors with 256 GB memory, we were not able to complete this classification task.

11.5 Flat Versus Hierarchical Classification Methods

In this section we investigate which classification approach is best suited to classify documents on the subgroup level of the IPC. Since we were not able to construct a flat classifier on subgroup level for the CLEF-IP 2010 data set, our analysis will be limited to the WIPO-alpha data set. In this section we will only consider unigram features; the relative merit of the different text representations will be discussed in Sect. 11.6. For the sake of comparison, we have included the most recently reported results, i.e. from [4], who also performed a subgroup classification on the WIPO-alpha set. It should, however, be noted that our train/test split differs slightly from theirs, which makes direct comparison impossible.

Table 11.5 summarises the success@rank scores for the odd numbers of the top 11 ranks of both the flat and hierarchical classifiers on the official test set of the WIPO-alpha corpus. The scores are calculated over the final rankings of 1140 subgroup category labels.

The results show that the flat and hierarchical classifiers achieve similar accuracy. We determined the significance of the differences from the confidence intervals: given the sample size, i.e. number of documents in the test set, the 95 % confidence interval for the success@rnk1 is ±0.95 % for both the flat and the hierarchical classifiers. We find that only from rank 9 onwards, the results do no longer fall in each other's confidence intervals, i.e. the differences are significant.

Our two-step classifier outperforms the two-step classifier of [4] by a large margin. With their additional third step, they reach a higher performance (36.1 %). However, since this result was obtained with a system that was tuned on the test set (see Sect. 11.2), it cannot be claimed that their three-phase method performs better than our two-step method. We will return to this finding in the discussion.

Unlike [9], we find similar performance for the flat and hierarchical approaches—at least until rank 9—while we had expected the hierarchical approach to outperform its flat counterpart: both approaches suffer from the same problem with sparse training material on subgroup level, but the flat classifier has a more

Table 11.5 Classification results of hierarchical and flat classifiers on subgroup level for WIPO-alpha test set using only unigram features

success@rnk	1	3	5	7	9	11
Hierarchical classification (%)	31.5	46.8	54.5	59.5	63.1	65.8
Flat classification (%)	31.8	46.6	53.9	57.9	61.0	63.6
Chen and Chang, two-step classification (%)	20.2					
Chen and Chang, with additional 3rd step (%)	36.1					

Table 11.6 success@rnk scores for subclass and subgroup classifiers in the hierarchical classifier on the WIPO-alpha set

success@rnk	1	3	5	7	9	11
Subclass classifier (%)	50.7	70.0	76.9	80.5	82.7	84.3
Chen and Chang subclass classifier[a] (%)	43.3	67.5	76.0	81.5	85.8	88.5

[a] Please note that these results are reported over a different test set (400 documents) and consequently are indicative for but not directly comparable to the other reported scores

Table 11.7 success@rnk1 scores for oracle runs on the WIPO-alpha set

Oracle runs	success@rnk1	Chance level
All subgroup categories	58.3 %	26.9 %[a]
Subgroup categories with 1 sister	87.0 %	50.0 %
Subgroup categories with 2 sisters	68.6 %	33.3 %
Subgroup categories with ≥3 sisters	56.3 %	25.0 %

[a] We calculated the micro-averaged chance level (in an oracle setting) by summing up the chance level of all documents (in the relevant subclass world) and then averaging over the number of documents

complex classification task (1140 vs. 11 categories on average for the subgroup classifiers in the hierarchical approach).

In the remainder of this section, we analyse the performance of the hierarchical classifier by analysing the performance of its individual components. First, we consider the subclass classifier on the first level in the hierarchy. This classifier achieved 50.7 % success@rnk1, which is similar to the state-of-the-art classification results on subclass level reported by Tikk et al. [28] and better than the subclass classifier of [4] (Table 11.6).

The 339 individual subgroup classifiers are trained on significantly less data than the subclass classifier on the first level. We evaluated these subgroup classifiers in 'oracle runs', i.e. runs in which the documents were only tested against subgroup models within the correct subclass world, effectively assuming a perfect classification on the first level in the hierarchy. The results of these experiments are given in Table 11.7. Please note that the last three lines show the performance of different sets of subgroup classifiers, grouped according to the number of daughters present in the subclass world.

In general, the subgroup classifiers seem to be of good quality and perform quite well (in an oracle setting). So given the good performance in smaller, contained worlds, how do we account for the relatively low accuracy (see Table 11.5) when the subgroup classifiers are used in the hierarchical setting where a document is scored by all subgroup classifiers?

First, there is the well-known problem of propagation of error: Table 11.6 shows that for 50 % of the test documents, the highest scoring subclass category is the correct one. For an additional 20 % of the test documents, the correct subclass category can be found at rank 2 or 3, while the correct labels of the remaining 30 % lie scattered at lower ranks. Given the difficulties in fitting sigmoids to

Table 11.8 Corpus statistics for subclass, class and section worlds in the WIPO-alpha training set after sample selection

	# of cat	av. size (stdev) in # doc	av. # daughters (stdev)	# of categories with one subgroup daughter
Subclass	339	38.00 (53.19)	3.36 (4.36)	128
Class	107	120.40 (179.48)	10.65 (15.08)	18
Section	8	1610.38 (945.85)	142.5 (77.8588)	0

subgroup classifier output (reported in Sect. 11.4.2.2), the probability estimates on the subgroup level may not be sufficiently powerful to repair the ‘errors’ made by the subclass classifiers.

Second, there are reasons for doubting whether classification at the subgroup level is at all feasible: Eisinger et al. [10] point out that in quite some cases patent documents should have additional labels on the subgroup level and that the labels that have been manually assigned by the patent examiners are to some extent arbitrary. Given the inconsistencies in the manually assigned labels on a level with fine-grained distinctions between categories, it is extremely unlikely that an automatic system can reproduce the manual labels with 100 % accuracy.

Third, our manner of training may have introduced an overlap between the class profiles[19]: the analysis of the class profiles of the subgroup categories in the flat and hierarchical classifiers shows that class profiles in the flat classifier generally contain more terms and, more specifically, they contain more ‘negative terms’. Terms with high negative Winnow weights characterise those unigram features that describe the rest of the corpus, not the category itself. They are especially useful in countering the positive weights of features that occur in many documents. Since the subgroup classifiers are trained in isolation, i.e. each in their own (small) subclass world with no information on the rest of the corpus, the models often do not contain enough negative terms to distinguish between categories in the testing phase.

The smaller number of negative terms (compared to positive terms) in the subgroup profiles for the hierarchical classifier indicates the lack of negative training material for the subgroup categories in the subclass worlds. Given the high number of single-daughter worlds (see Table 11.1), this is not surprising. We therefore hypothesised that training in larger contexts is better for optimal performance in a hierarchical system. To examine this hypothesis, we performed additional experiments in which subgroup classifiers were trained in larger ‘worlds’, i.e. the classifiers for individual subgroup categories were trained against all other subgroup categories in the same class (C) or section (S) in the IPC hierarchy. Table 11.8 shows corpus statistics on these larger worlds.

Although the different data selection criteria result in larger class profiles (with a higher ratio of negative terms compared to positive terms), Table 11.9 only shows

[19] Class profiles are the category models which comprise the most relevant terms for each category with their corresponding Winnow weights.

Table 11.9 Classification results for WIPO-alpha test set after different training data selection

success@rnk	1	3	5	7	9	11
Trained on SC world (%)	31.5	46.8	54.5	59.5	63.1	65.8
Trained on C world (%)	32.0	47.4	55.1	60.0	63.3	66.1
Trained on S world (%)	32.1	48.4	55.9	60.5	63.8	66.4

The first row is the same as the first row of Table 11.5

marginal and non-significant improvements between the different runs. Analysis of the class profiles of the categories trained in the *class* and *section* worlds shows that the added terms tend to have low Winnow weights and have relatively little impact on classification performance.

So, even with more negative training data, the hierarchical classifier does not rise above the performance level of the flat classifier. We must conclude that the overlap between the categories on the lowest levels and the small number of training documents in many 'worlds' are an insurmountable problem in the WIPO-alpha training/test set.

It might be argued that the classification at subgroup level should not be approached by means of a classifier that relies on some kind of training, simply because of the lack of sufficient amounts of training data. Chen and Chang [4] obtained a substantial improvement on the subgroup level by using a KNN classifier. We conducted a large number of experiments in which we used the features selected for the Winnow classifier in two different KNN classifiers, TiMBL [5] and sklearn [21]. However, we were not able to obtain a better classification accuracy than with the Balanced Winnow algorithm.

As mentioned above, the WIPO-alpha set is hardly representative of a real-life task. The CLEF-IP 2010 corpus is much larger, both in the number of documents and in the number of categories that must be distinguished on subclass and subgroup levels. While flat classification on such a set is not feasible for our classification algorithm, we expect that the hierarchical approach, which is much more scalable, will yield similar results (as a hypothetical flat one), since that was the case for the WIPO-alpha corpus. Furthermore, the larger amount of data opens possibilities to examine the impact of more precise text representations, which might help to solve the problem of the high overlap between the subgroup categories.

11.6 The Impact of Phrasal Features

In this section we examine the impact of different text representations on classification accuracy for different levels of the IPC hierarchy. For this series of experiments, we used the CLEF-IP 2010 corpus in addition to the WIPO-alpha data set, since the data sparseness in the WIPO-alpha corpus is especially problematic for the inherently sparse skipgram features. Furthermore, the CLEF-IP 2010 corpus is much more representative for the patent classification task than the WIPO-alpha benchmark, both in terms of the number of documents and the number of categories available.

Our goal is twofold: (1) We will examine the (relative) improvements of adding skipgrams for the subclass and subgroup classifiers. Our hypothesis is that on the subgroup level, in which the categories tend to overlap more, the more precise distinctions provided by the phrasal features will have a larger impact than on the subclass level. (2) We will compare the effects of adding features for both the CLEF-IP 2010 and the WIPO-alpha set in order to obtain a better understanding of how much training material is needed for phrasal features to be effective. It should be noted that in this section we use a different evaluation measure than in the previous section: up to now we have reported success@rnk for the sake of comparison with [4]. Since the CLEF-IP 2010 set is a multi-label set with a varying number of relevant categories per document, this measure is no longer adequate. We will therefore report our results using the well-known precision, recall and F1-measures. Relevant output rankings from classification experiments discussed in the previous section have been (re-)evaluated using these metrics.[20]

As is shown in Table 11.2, our train/test split for the CLEF-IP 2010 corpus consists of 595,080 and 396,725 documents, respectively, with 575 categories on subclass level and 19,441 on subgroup level. Unlike the WIPO-alpha documents, each document in the CLEF-IP 2010 set may have multiple relevant category labels. In the case of multi-label classification, the LCS can return a varying number of categories per document. This is determined by three parameters: (1) a threshold that puts a lower bound on the classification score (in this case probability) for a class to be selected, (2) the maximum number of classes selected per document ('maxranks') and (3) the minimum number of classes selected per document ('minranks'). Setting minranks $= 1$ assures that each document is assigned at least one category, even if all categories have a score or probability below the threshold. We used the cross-validation folds to determine the optimal evaluation configuration, which resulted in the following setting: minranks $= 1$, threshold $= 0.8$ and maxranks $= 8$ and 20 for the subclass and subgroup classifiers, respectively.

First we study the impact of adding skipgrams on subclass level. Table 11.10 shows the precision, recall and F1 scores for the CLEF-IP 2010 test set (left-hand side) and WIPO-alpha data set (right-hand side), respectively. Please note that these scores cannot be directly compared as they are (a) based on different data sets with a different number of categories to be distinguished and (b) a substantially different classification problem: classifying the WIPO-alpha set is a mono-label classification task, while the CLEF-IP 2010 set is multi-label. The scores should rather be seen as an indication of the difficulty of classifying on a certain level in the IPC hierarchy.

For both the WIPO-alpha and CLEF-IP 2010 test sets, we can see an improvement of classification performance on subclass level when skipgrams are added.

[20] Since we defined the classification task on the WIPO-alpha set as a mono-label task where the classifier *must* return one label, the reported (micro-averaged) scores will always yield equal precision and recall scores.

Table 11.10 Classification results of unigrams-only and unigrams+skipgrams classifiers on subclass level for the CLEF-IP 2010 corpus and the WIPO-alpha corpus

	CLEF-IP			WIPO-alpha
	P	R	F1	P = R = F1
Unigrams (%)	63.9	62.3	63.1	50.7
Unigrams + skipgrams (%)	66.6	67.3	66.9	51.9

Table 11.11 Classification results of unigrams-only and unigrams+skipgrams hierarchical classifiers on subgroup level for the CLEF-IP 2010 corpus and the WIPO-alpha corpus

	CLEF-IP			WIPO-alpha
	P	R	F1	P = R = F1
Unigrams (%)	45.1	27.7	34.3	31.5
Unigrams + skipgrams (%)	52.7	30.3	38.4	32.5

We determined the significance of the differences between the unigrams and unigrams+skipgrams using the confidence intervals: Given the sample sizes, i.e. the number of documents in the respective test sets, the 95 % confidence interval for the F1 values is ±0.15 % and ±1.02 % for the CLEF-IP 2010 and the WIPO-alpha subclass classifiers, respectively. From this we can conclude that adding skipgrams leads to a significant improvement in the CLEF-IP 2010 set, but not in the WIPO-alpha set. As there is much more training material per category available in the CLEF-IP 2010 data set, compared to WIPO-alpha data set, the inherently sparse skipgram features attain high enough frequencies to aid in the classification process.

Table 11.11 shows the results for the subgroup rankings of the hierarchical classifiers, also for the CLEF-IP 2010 and WIPO-alpha test sets.

Here too we find a significant improvement for the combined run for the CLEF-IP 2010 set, but not for the WIPO-alpha set (with confidence intervals of ±0.15 % and ±0.95 % for the F1 scores of the CLEF-IP 2010 and WIPO-alpha set, respectively).

If we compare the (relative) improvements in F1 scores of the combined runs with the unigram runs for both the CLEF-IP subclass and subgroup classifiers, we find a similar improvement (around 4 percentage points) on both levels. We can therefore conclude that combining unigrams and skipgrams is beneficial for classification performance on any level in the IPC hierarchy. However, our initial hypothesis that skipgrams would have a larger impact on lower—and supposedly more overlapping—levels in the hierarchy is not confirmed. Close analysis of the class profiles does reveal that on average skipgrams occur at higher ranks in the subgroup class profiles than in the subclass class profiles. It seems that these features fill up the feature space when the unigram features are not sufficiently discriminative. Therefore, the hypothesis that skipgrams are more important on subgroup than on subclass level cannot be rejected either. It may be that even the CLEF-IP corpus is too small to allow for a decisive test.

As regards the second research question, the relative improvements between the classification results for the CLEF-IP 2010 and the WIPO-alpha sets clearly confirm our hypothesis that adding phrasal features is only effective when enough training data is available. For the CLEF-IP set, the skipgrams lead to a highly significant improvement on subgroup level, despite the fact that there is much less training material available than on subclass level. This suggests that an average number of 142 training documents per category are enough training data to see an impact of skipgram features, despite the skewed distribution of the number of documents per category.

11.7 Conclusion

In this chapter we examined the feasibility of performing classification on subgroup level of the IPC taxonomy. This task is generally considered extremely difficult because of three problems reported in the literature: (a) The overlap between categories is too large, and differences are too subtle to be captured adequately. (b) The number of categories is exceedingly large, which leads to scalability issues. (c) The data sparseness (in the number of documents per category) at the lowest level is too severe to build adequate classification models.

In our research we focused on two main questions which address these difficulties: (1) Can we circumvent the problems of overlap and the number of categories by using a hierarchical approach to classification on subgroup level and how does it compare to a flat classification approach? (2) Can we improve the classification on subgroup level by adding phrasal features, namely, skipgrams, to unigram features and how does the impact correlate with the granularity of the different levels in the IPC hierarchy? We performed classification experiments on the WIPO-alpha benchmark set, as well as on the much larger and more realistic CLEF-IP 2010 data set.

Our hierarchical approach consisted of a two-step top-down classification system with a subclass classifier on the top level and a set of subgroup classifiers—each trained within a subclass world—on the lower tier. The scores of the individual classifiers were converted to probabilities, which were then combined in a weighted scheme. To minimise the propagation of error and effectively allow high-scoring subgroup categories to move up in the final ranking, we did not define any cut-off thresholds on the subclass level during the testing process.

Regarding the first research question, we found that the flat and hierarchical approaches achieve similar accuracy scores on the WIPO-alpha set (31.5 % and 31.8 % success@rnk1, respectively). This shows that when it becomes infeasible to train a flat (text) classifier because the number of categories that must be distinguished is too large, a hierarchical classifier might be a good alternative for classification on the lowest level(s) of a taxonomy. Using a hierarchical approach, we were able to transform a 19,441-category problem into smaller, manageable subproblems and perform subgroup classification for a 900K corpus with encouraging accuracy.

Regarding the second research question, we were able to replicate the improvements of combining unigrams with skipgrams which were previously observed in [6, 7]. We did not observe a difference in the effect size of adding skipgrams to unigrams between the different IPC levels.

The difference in size between the two WIPO-alpha and CLEF-IP corpora gave us insight into the problems caused by data sparseness on subgroup level. Our best-scoring approach (hierarchical approach with unigram+skipgram features) achieved 32.5 % F1 accuracy for subgroup classification on the WIPO-alpha set (mono-label) and 38.4 % on the CLEF-IP 2010 set (multi-label). Since skipgrams are inherently sparse, a sufficiently large amount of training data must be available before phrasal features attain high enough frequencies to aid in the classification process. We found that—for classification on subgroup level in the CLEF-IP 2010 set—an average of 142 documents per category was enough to see a significant impact of adding skipgram features. We conjecture that with less training material available, case-based methods such as KNN might be preferred for classification on the lowest levels of the IPC taxonomy, even if our attempts to use KNN-based subgroup classifiers in WIPO-alpha data set were not successful.

An interesting pattern that we observed in both the WIPO-alpha and CLEF-IP hierarchical classifiers was the low weight given to the subgroup probabilities in the weighting of the probability estimates to reach optimal ranking. The fact that this occurs independent of the amount of training data available—as described in Sect. 11.3.1.2 we took care to avoid data sparseness problems when selecting a subset from the CLEF-IP data set—seems a strong indication that no matter how much training material is available, (model-based) classification on the subgroup level is a hazardous undertaking. We suspect, however, that the small numbers of documents in some subgroups are less of a problem than the reliability and completeness of the manually assigned labels, which serve both for supervising the training and as a reference in the evaluation of the classifier output.

As a final recommendation for future work in the patent classification field, we would like to promote the use of the CLEF-IP sets as future benchmarks: while we found that the WIPO-alpha set is a clean and usable data set, the CLEF-IP data set presents a more realistic task,[21] both in the number of categories and in the amount of training data available. Especially the latter is of great importance for further research focusing on (sub)group levels.

Acknowledgements This chapter is dedicated to the memory of Kees Koster, who was a major influence on this research.

[21] An even more realistic data set, called DOCDB, is hosted at the EPO but is not freely available.

References

1. Benzineb K, Guyot J (2011) Automated patent classification. In: Current challenges in patent information retrieval. Springer, New York, pp 239–261
2. Cai L, Hofmann T (2004) Hierarchical document categorization with support vector machines. In: Proceedings of the thirteenth ACM international conference on information and knowledge management, CIKM '04. ACM, New York, pp 78–87
3. Cesa-Bianchi N, Gentile C, Zaniboni L (2006) Incremental algorithms for hierarchical classification. J Mach Learn Res 7:31–54
4. Chen YL, Chang YC (2012) A three-phase method for patent classification. Inf Process Manag 48(6):1017–1030
5. Daelemans W, Zavrel J, van der Sloot K, van den Bosch A (2010) TiMBL: Tilburg memory-based learner - version 6.3 - Reference Guide
6. D'hondt E, Verberne S, Weber N, Koster K, Boves L (2012) Using skipgrams and pos-based feature selection for patent classification. Comput Linguist Neth J 2:52–70
7. D'hondt E, Verberne S, Koster K, Boves L (2013) Text representations for patent classification. Comput Linguist 39(3):755–775
8. D'hondt E, Verberne S, Oostdijk N, Beney J, Koster C, Boves L (2014) Dealing with temporal variation in patent categorization. Inf Retr. doi:10.1007s10791-014-9239-6
9. Dumais S, Chen H (2000) Hierarchical classification of web content. In: Proceedings of the 23rd annual international ACM SIGIR conference on research and development in information retrieval, SIGIR '00. ACM, New York, pp 256–263
10. Eisinger D, Tsatsaronis G, Bundschus M, Wieneke U, Schroeder M (2013) Automated patent categorization and guided patent search using IPC as inspired by MeSH and PubMed. J Biomed Semant 4(1):1–23
11. Fall CJ, Benzineb K (2002) Literature survey: issues to be considered in the automatic classification of patents, pp 1–64
12. Fall CF, Benzineb K, Guyot J, Törcsvári A, Fiévet P (2003) Computer-assisted categorization of patent documents in the international patent classification. In: Proceedings of the international chemical information conference
13. Falquet G, Guyot J, Benzineb K (2010) myClass: a mature tool for patent classification. In: Multilingual and multimodal information access evaluation - proceedings international conference of the cross-language evaluation forum, CLEF 2010. Springer, Berlin
14. Guyot J, Benzineb K (2013) IPCCAT-report on a classification test. Tech. Rep., Simple Shift. srv1.olanto.org/download/myCLASS/publication/IPCCAT_Classification_at_Group_Level_20130712.pdf
15. King G, Zeng L (2001) Logistic regression in rare events data. Polit Anal 9(2):137–163
16. Koster CH, Beney J, Verberne S, Vogel M (2010) Phrase-based document categorization. Springer, New York, pp 263–286
17. Krier M, Zaccà F (2002) Automatic categorization applications at the European patent office. World Patent Inf 24(3):187–196
18. Li Y, Bontcheva K, Cunningham H (2007) Svm based learning system for f-term patent classification. In: Proceedings of the 6th NTCIR workshop meeting on evaluation of information access technologies: information retrieval, question answering and cross-lingual information access (NTCIR'07), pp 396–402
19. Lin HT, Lin CJ, Weng RC (2007) A note on Platt's probabilistic outputs for support vector machines. Mach Learn 68(3):267–276
20. Oostdijk N, Verberne S, Koster C (2010) Constructing a broadcoverage lexicon for text mining in the patent domain. In: Proceedings of the international conference on language resources and evaluation, LREC 2010, 17–23 May 2010, Valletta, Malta

21. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E (2011) Scikit-learn: machine learning in Python. J Mach Learn Res 12:2825–2830
22. Piroi F, Lupu M, Hanbury A, Sexton AP, Magdy W, Filippov IV (2010) CLEF-IP 2010: retrieval experiments in the intellectual property domain. In: Proceedings of CLEF 2010 (notebook papers/labs/workshops)
23. Piroi F, Lupu M, Hanbury A, Zenz V (2011) CLEF-IP 2011: retrieval in the intellectual property domain. In: Petras V, Forner P, Clough PD (ed) Proceedings of CLEF 2011 (notebook papers/labs/workshop)
24. Platt J (1999) Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In: Advances in large margin classifiers. MIT Press, Cambridge, MA, pp 61–74
25. Salton G, Buckley C (1988) Term-weighting approaches in automatic text retrieval. Inf Process Manag 24:513–523
26. Silla C, Freitas A (2011) A survey of hierarchical classification across different application domains. Data Min Knowl Disc 22(1–2):31–72
27. Smith H (2002) Automation of patent classification. World Patent Inf 24(4):269–271
28. Tikk D, Biró G, Törcsvári A (2007) A hierarchical online classifier for patent categorization. IGI Global, Information Science Reference, Hershey, pp 244–267
29. van Halteren H (2000) The detection of inconsistency in manually tagged text. In: Proceedings of LINC-00
30. Wang X, Zhao H, Lu BL (2011) Enhance top-down method with meta-classification for very large-scale hierarchical classification. In: Proceedings of the international joint conference on natural language processing, pp 1089–1097
31. Widodo A (2011) Clustering patent documents in the field of ICT (information and communication technology). In: Proceedings of 2011 international conference on semantic technology and information retrieval (STAIR), pp 203–208