applied sciences

MDPI

Article

On the Potential of Taxonomic Graphs to Improve Applicability and Performance for the Classification of Biomedical Patents

Kai Frerich [1], Mark Bukowski [1], Sandra Geisler [2] and Robert Farkas [1,*]

[1] Department of Science Management, Institute of Applied Medical Engineering, RWTH Aachen University—University Hospital Aachen, 52074 Aachen, Germany; kai.frerich@rwth-aachen.de (K.F.); bukowski@ame.rwth-aachen.de (M.B.)
[2] Fraunhofer Institute for Applied Information Technology FIT, Schloss Birlinghoven, 53757 Sankt Augustin, Germany; sandra.geisler@fit.fraunhofer.de
* Correspondence: farkas@ame.rwth-aachen.de; Tel.: +49-241-80-80740

Abstract: A core task in technology management in biomedical engineering and beyond is the classification of patents into domain-specific categories, increasingly automated by machine learning, with the fuzzy language of patents causing particular problems. Striving for higher classification performance, increasingly complex models have been developed, based not only on text but also on a wealth of distinct (meta) data and methods. However, this makes it difficult to access and integrate data and to fuse distinct predictions. Although the already established Cooperate Patent Classification (CPC) offers a plethora of information, it is rarely used in automated patent categorization. Thus, we combine taxonomic and textual information to an ensemble classification system comparing stacking and fixed combination rules as fusion methods. Various classifiers are trained on title/abstract and on both the CPC and IPC (International Patent Classification) assignments of 1230 patents covering six categories of future biomedical innovation. The taxonomies are modeled as tree graphs, parsed and transformed by Dissimilarity Space Embedding (DSE) to real-valued vectors. The classifier ensemble tops the basic performance by nearly 10 points to F1 = 78.7% when stacked with a feed-forward Artificial Neural Network (ANN). Taxonomic base classifiers perform nearly as well as the text-based learners. Moreover, an ensemble only of CPC and IPC learners reaches F1 = 71.2% as fully language independent and straightforward approach of established algorithms and readily available integrated data enabling new possibilities for technology management.

Keywords: innovation management; medical technology; taxonomies; tree edit distance; multiclass patent categorization; automation; emerging technologies

Citation: Frerich, K.; Bukowski, M.; Geisler, S.; Farkas, R. On the Potential of Taxonomic Graphs to Improve Applicability and Performance for the Classification of Biomedical Patents. *Appl. Sci.* **2021**, *11*, 690. https://doi.org/10.3390/app11020690

Received: 30 November 2020
Accepted: 8 January 2021
Published: 12 January 2021

1. Introduction

The analysis of patents is one of the core duties of technology and innovation management with varying purposes and perspectives, such as forecasting emerging technologies, assessing performances of regional/national innovation systems, mapping technologies, managing R&D activities, or evaluating the collaboration potential at company or policy level [1,2].

An essential subtask within these processes is classifying patents into coherent groups of similar documents as base for further retrieval and assessment. To facilitate such approaches, examiners at the national authorities use official taxonomies to assign patent applications according to their content to one or more classes. To overcome the so far existing separation, in 2013 the European Patent Office (EPO) and the United States Patent Office (USTPO) jointly released the Cooperative Patent Classification (CPC), which compared to the previously used International Patent Classification (IPC) now comprises about four times the number of different classes [3]. However, in order to assess domain-specific emerging technologies, either the use of concordance tools/tables to map, e.g., medical

technologies [4] or a complete reclassification of patents beyond the official taxonomy systems is still indispensable [1].

Facing the huge and still expanding number of worldwide patents [5], the implementation of automated text categorization methods becomes vital to keep up with the rapidly growing body of knowledge. Many approaches have been deployed differing in features, such as Machine Learning (ML) algorithms, domains etc. Most of these approaches use the bag-of-word model to transform the text into term frequencies as real-valued features for machine learning (Term Frequency-Inverted Document Frequency—TF-IDF) [6]. In order to improve the classification performance, advanced methods incorporate more and more (meta) data of several distinct sources [7].

However, the subsequently increasing complexity raises new difficulties in mastering both varying methods and different data sources, such as data accessibility and integration (deduplication), the number of features in sparse matrices, the need of a method to combine multiple predictions, or the necessary computing power. Surprisingly, only a few studies implement IPC, and even less use the CPC assignments as an additional information source, although the data are readily available jointly with patent texts.

Furthermore, the well-studied fusion of different classifiers promises improved performance results especially when the diversity of the base classifiers is high [8]. A plethora of fusion methods, e.g., rule-based approaches such as summing and averaging, or stacking with machine leaning algorithms acting as fusion classifiers, are available to customize ensemble classification systems to specific tasks.

Despite of these advantages, so far patent categorization into user-defined groups via ensemble classifier systems has rarely been studied. In particular, the use of the taxonomies mentioned above could be a major step to overcome given limitations, since IPC and CPC represent a graph that provides available and extensive information. However, to be used in a classifier ensemble, the graph representation must be transformed into a real-valued vector. Riesen and Bunke [9] offer an outstanding solution for this challenging task with the Dissimilarity Space Embedding (DSE), which provides a dense real-valued vector of graph edit distances.

Thus, we present a novel ensemble classification system based on the combination of text and taxonomic features to enhance both applicability and performance in a real-world set-up, namely the mapping of patents as an important R&D output of biomedical engineering into six fields of future innovation (e.g., telemedicine, imaging, and implants). The central question is to what extent classification performance can be improved without significantly increasing the complexity of the approach.

To achieve this, as far as possible robust technologies and models and only one source providing integrated data will build up our approach. In detail, title and abstract of biomedical patents are processed according to the bag-of-words model. Additionally, the IPC and especially the much more detailed CPC are transformed by means of DSE, an established method that—to the best of our knowledge—is now applied to patent taxonomies for the first time. Both textual and taxonomic features serve as input to four different machine learning base classifiers to assign patents into six classes of future biomedical innovation. By pairing two base learners applied on diverse features (one text- and one taxonomy-based) selected according to the differing performances on validated test data (low vs. top performer) and varying the fusion methods systematically between stacking or fixed combination rules, our extensive comparisons totals in 64 different ensembles. The results show at first that IPC and CPC related learners achieve comparable performances as those using textual features in base classification and secondly that the top taxonomic base classifiers contribute substantially to the overall performance when stacked with also top performing base learners on textual features.

The remainder of this paper ranges from analyzing of the related work, to describing materials and methods ending up at presenting and discussing the results. The conclusions point out the major findings and important future perspectives.

2. Related Work

2.1. Official Patent Classification Systems

The IPC system with approximately 70,000 classes and its extension, the unified CPC system with about 250,000 classes, are currently in use at the different patent authorities. As part of any patent application process, the examiners at the patent offices assign up to several hundred IPC/CPC codes to every filed patent to describe its content [3]. This independent assessment of the content makes both codes with their availability and language independence extremely valuable for classification tasks. While the first three levels of CPC and IPC (section, class, subclass) show a high degree of similarity, the greater level of detail provided by CPC is particularly evident from the group level and below [3]. Despite this new wealth of information, in general only a reserved scientific discussion of the CPC system has been conducted. Since 2013, about four times more publications listed in Web of Science deal with IPC than with CPC.

2.2. Automated Text Categorization using Patents

Classification algorithms are successfully applied to patent texts. In early approaches full texts serve as basis for automated categorization using k-Nearest-Neighbor (kNN), naïve Bayes, Support Vector Machine (SVM) or back-propagation neural network [10,11]. In order to improve the classification performance, which is reduced, inter alia, by the legal and blurred language of patents, further data from distinct sources were combined into increasingly complex models. For example, Liu and Shih [7] suggest a hybrid classification with a weighted linear combination of content-, citation-, metadata-, and network-based predictions, which outperformed each single contribution with F1 = 86.4%. However, this performance is contrasted by an enormous complexity and effort: kNN and SVM, cosine similarity, ontology-based network analysis, more than 27,000 patents for training purposes, and the adjustment of weights for the final prediction. All this illustrates the methodological challenges and the workload, which have to be overcome.

In contrast, with only 1600 training objects a multiclass setting can successfully be established using SVM (66.4% accuracy) leading the classifier ranking followed by random forest and kNN [12]. Furthermore, SVM served as both base and fusion classifier in an ensemble approach and achieved F1 = 77.7%, further increased by implementing user feedback as active learning (F1 = 84.2%) [13]. In order to achieve these high performances in the chosen multiclass classification, Zhang interactively extended his model not only to active learning, but also by reducing the text features using principal component analysis or reinforcing the training process through so-called *dynamic certainty propagation*.

IPC or CPC have rarely been used as a knowledge source, but rather as a target for automated document categorization to relieve the examiners from manual work [7,11,14]. Recent deep learning approaches such as BERT as well proved their predictive power in mastering the assignment of patents at the CPC subclass level [15]. In general, this evolving number of approaches with extensive pre-trained language models has a great potential for semantic tasks such as text classification. However, a high complexity (e.g., in terms of fine-tuning) and demanding huge amount of training data limit the applicability to classify patents, especially into user-defined categories with usually very limited amount of training samples. Therefore, our approach focuses on applicability with established methods as a solid basis for a proof of concept. This could serve as a basis for future studies to evaluate whether more complex methods such as BERT are worthwhile in terms of effort and outcome.

2.3. Ensemble Classification

There is an extended body of research and general knowledge on combining classifiers [16–18] to achieve, e.g., an improvement of the overall performance [8,19] despite of noisy training data. Especially the diversity provided by the base classifiers bolsters the capability of their fusion [20]. Among different options to introduce this diversity,

e.g., varying the base ML classifier, the use of distinct features extracted from the same data are considered to be the most promising approach [8].

The final performance depends on the scheme of combining the basic predictions. Both methods, fixed combinations rules (FCR), i.e., combining the estimates of posterior class probability using algebraic operations such as summation or product [21], and stacking, i.e., using a classifier to fuse the probability estimates of the base learners [22], are successfully deployed in text classification. Surprisingly, patents are very rarely the subject of classifier combinations, even though, unlike scientific texts, patents are considered particularly difficult to classify. This is true even when considering boosting, a technique in which a larger number of weak learners are successfully combined to achieve a strong ensemble performance. During the last 25 years, especially adaptive boosting (AdaBoost) [23], providing increased weights during training to the base learners with higher predictive power, has been studied intensively. AdaBoost works especially well on weak learners performing only slightly better than a random classification, whereas using stronger base classifiers such as SVM does not lead to improved results [24]. More recently, Lee uses a combination of topic modeling and AdaBoost to predict the suitability of patents for further technology transfer (Lee et al. 2018) and achieves a maximum F1 value of approx. 0.589. Apart from patent classification, currently AdaBoost with SVM is, e.g., successfully applied to issues such as imbalanced streaming classification and concept drift [25,26].

2.4. Feature Extraction from Graphs

The IPC and even more the CPC taxonomy constitutes a powerful representation of information regarding the content of the underlying patents, structured as a graph of nodes and edges. A plethora of research work has been performed on studying graph-based pattern representation [27]. To utilize patent taxonomies as input for machine learning classification the graph representation (G) must be transformed into a real-valued feature domain.

Besides extracting structural information of graphs with, e.g., the adjacency matrix or the Laplacian matrix methods [28], the widely used Graph Edit Distance (GED) approach compares two graphs by computing the distance as a cost function, i.e., on a minimized set of edit operations such as inserting, deleting, and substituting nodes or edges to transform one graph into the other. In their pioneering work on Vector Space Embedding, Riesen and Bunke [9] introduced the concept of a variety of selected prototypes, i.e., training graphs to which the GED of an input graph are computed. The resulting dissimilarity representation characterizes the specificity of the input graph as dense real-valued vector, which is well suited as input for the subsequent classification algorithms. Additionally, different graph edit approximations are successfully deployed even in an ensemble setting outperforming the best individual base classifier [29].

3. Materials and Methods

A classifier C solving a supervised learning problem for a given set of n classes $\Omega = \{\omega_1, \ldots, \omega_n\}$ is described by a set of functions $\{c_1, \ldots, c_n\}$ defined as $c_i : \mathcal{X} \to [0, 1]$. Here c_i assigns every object from the input space $\mathcal{X}$ a non-negative score for each class ω_i. Additionally, we expect the sum of all class scores to be equal to 1. Thus, the used classifiers are called probabilistic because their score outputs can be interpreted as the likelihood of an object $x \in \mathcal{X}$ belonging to the class ω_i. To combine the strengths of different approaches to automatically assign patents to biomedical classes of future innovation a unified ensemble classification system is created as follows (see Figure 1):

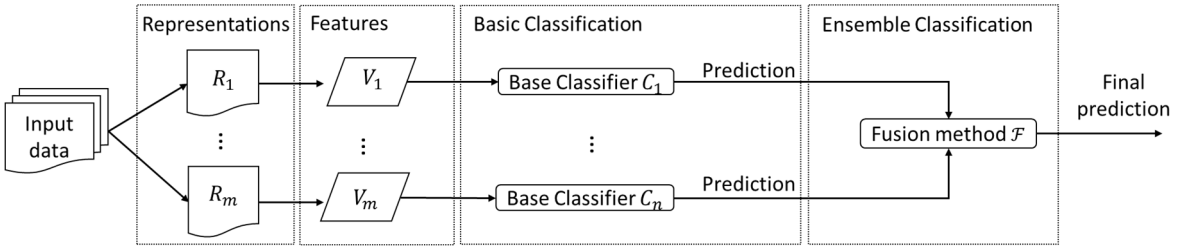

Figure 1. General structure of a classification model using different features V of representations R of input data for an ensemble of n base classifiers C merged by a fusion method F.

In our approach, at the input site two different representations of each patent document, namely text (title, abstract) and assigned taxonomies (CPC and IPC), are transformed into numerical features to be fed to machine learning base classifiers. After hyperparameter tuning, four different base classifiers compute their predictions from distinct features of the same object (Stage I) to be finally merged by various fusion methods providing the final prediction (Stage II). The whole two-stage-process (see Figure 2) has been implemented using Python and the scikit-learn library [30].

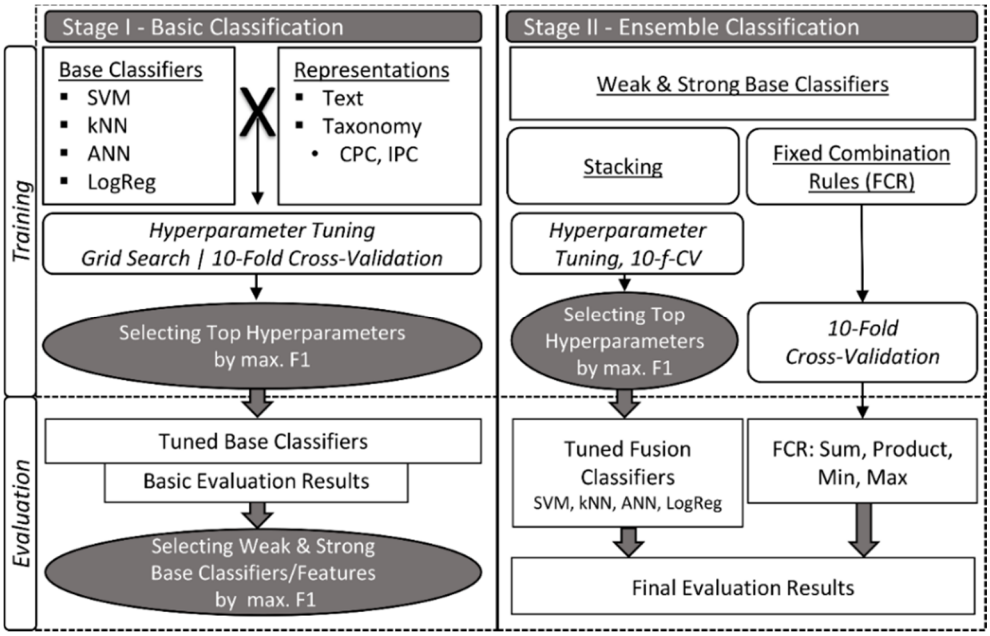

Figure 2. Overview of the main stages of data processing and evaluation. SVM—Support Vector Machine, ANN—Artificial Neural Network, kNN—k-Nearest-Neighbor, LogReg—Logistic Regression.

3.1. Patent Data

The training data consists of 1230 distinct patents, which are equally assigned to six classes of innovative biomedical devices (see Table 1). This dataset is based on our preliminary work [31]: the data were selected and labeled with a study-based keyword search and further optimized by experts. To evaluate the classification performance, an externally validated test data set is used comprising additional 94 patents, which were categorized by biomedical experts. All content of the patents, the text of titles and abstracts as well as the de-duplicated IPC and CPC codes were extracted from the patent database PATSTAT [32] and then transferred to feature transformation.

Table 1. Patent data for training and testing assigned into six classes of innovative biomedical devices.

Class Name	# of Patents		# of CPC Codes		# of IPC Codes	
	Training	Testing	Training	Testing	Training	Testing
Imaging	205	30	1380	129	838	84
Implants and prostheses	205	15	1659	78	732	33
Telemedicine	205	2	1064	5	827	6
Surgical intervention	205	23	1351	98	805	59
In-vitro diagnostics	205	6	922	18	1175	18
Special therapy systems	205	18	1190	67	1024	50
Total	1230	94	7566	395	5401	250

3.2. Feature Transformation

3.2.1. Textual Data

To obtain numerical feature vectors from textual data, the widely used 'bag-of-words' model is applied after preprocessing the concatenated title and abstract by removing stop-words not providing any additional information. Final feature vectors for text categorization are created utilizing a TF-IDF value computation for each input term per text document [6].

3.2.2. Taxonomic Data

By implementing the following steps, the graph based taxonomic data of IPC and CPC are transformed into a real-valued vector to be used as input for machine learning algorithms.

3.2.3. Tree Creation

Since both used patent classifications comprise a hierarchically organized taxonomy, each can be represented by a tree. By additionally inserting a root node, the trees of all assigned class codes are unified in one tree structured graph (see Figure 3), which facilitates the computing of the distance between the structured codes of CPC/IPC taxonomy using the tree-edit-distance [33].

Table 2. Excerpt from the CPC definition showing all hierarchical levels up to the actual code A61B5/0055. Symbols in square brackets indicate subgroups discarded by parsing the code string.

Level	Symbol	Classification and Description
Section	A	Human necessities
Class	A61	Medical or veterinary science; hygiene
Subclass	A61B	diagnosis; surgery; identification
Main Group	A61B 5/00	Detecting, measuring or recording for diagnostic purposes; identification of persons
Subgroup	(A61B 5/00 48)	• Detecting, measuring or recording by applying mechanical forces or stimuli
Subgroup	A61B 5/00 55	• • by applying suction

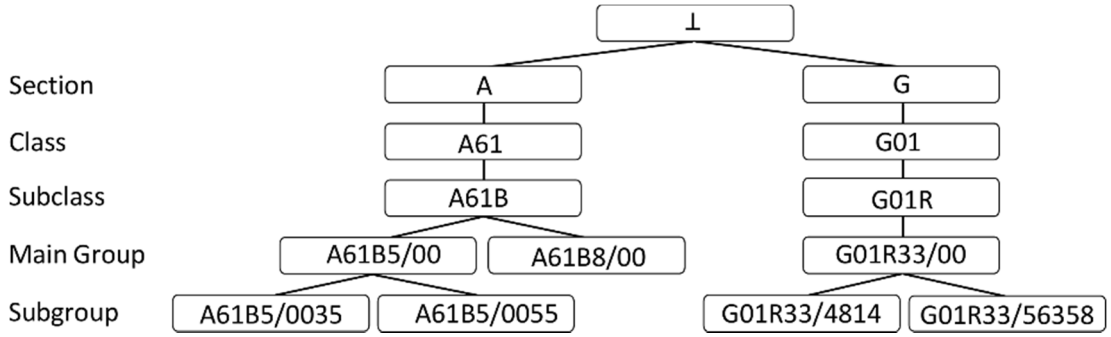

Figure 3. Example of a simplified Cooperative Patent Classification (CPC) tree of a patent parsed from the CPC code strings (see also Table 2).

The string representations of the IPC/CPC codes have to be parsed to build the required tree. However, a full deduction of the hierarchy as given in official CPC definition [34] solely from the code strings is not possible, because the different subgroup levels are not represented in the code strings. Thus, the resulting tree parsed from the code string is only an approximation of the actual defined structure, which is created by discarding eventually existing intermediate subgroup levels between the given code and the main group. For example, a patent is marked with code A61B5/0055. Since this is according to the official CPC definition a 2nd-level subgroup (see Table 2), the string parser discards one intermediate subgroup. Nevertheless, the parsed tree carries as much information as possible to be extracted solely from the string representation of a CPC/IPC code.

3.2.4. Vector Space Embedding

We adopt the *Dissimilarity Space Embedding* method (DSE) to transform the CPC and IPC tree into a vector space following the work of Riesen and Bunke [9]. A graph object can therefore be described by its dissimilarity to a fixed set of other objects from the same domain (prototypes).

Given a graph domain $\mathcal{G}$, a dissimilarity between graphs can be expressed using a distance function $d : \mathcal{G} \times \mathcal{G} \to \mathbb{R}^{\geq 0}$. Given such a distance function d, a dissimilarity space embedding γ can be defined as:

$$\gamma : \mathcal{G} \to \mathbb{R}^n,\ g \mapsto (d(g_1, g),\ \ldots, d(g_n, g)) \tag{1}$$

A suitable distance function in a dissimilarity space embedding is provided by the graph edit distance (GED), given by the minimal cost sequence of all operations transforming the graph g_1 into g_2 using insertion, deletion, and relabeling of nodes and edges with the cost of each operation set to 1. For computing the general GED no efficient algorithm is known [35]. However, since IPC/CPC can be represented as ordered trees, the problem can be reduced to compute the tree edit distance utilizing the Zhang-Shasha algorithm [36]. Although being an efficient algorithm, no implementation of sufficient performance was available. Thus, we implemented the algorithm in rust GitHub repository of 'Tree edit distance algorithm implemented in rust': https://github.com/AME-SCM/tree-edit-distance.

3.2.5. Prototype Selection Methods

Before any features are created invoking the function γ, a set of graphs $g_1, \ldots, g_n$ called 'prototypes' needs to be fixed. Prototypes are chosen from the training data as the source of known data in advance and are not altered thereafter. Two selection methods with different capabilities [9] were applied to optimize the resulting real-valued feature vectors: (1) *random prototype selector*—selecting prototypes in a completely random manner, and (2) *spanning prototype selector*—selecting the median graph as first element of the training data. The further added graphs maximize the distance to the nearest graph of the already

chosen one yielding a better representation of the graph domain. Hyperparameter tuning determines the number of prototypes needed to deliver meaningful features.

3.3. Experimental Settings

3.3.1. Classifier Selection and Hyperparameter Tuning

A set of four well established machine learning algorithms are selected particularly to enable comparisons to prior work: Support Vector Machine (SVM), k-Nearest Neighbor (kNN), Logistic Regression (LogReg), and feed-forward Artificial Neural Network (ANN). They all serve as both, base and fusion classifiers (see Figure 2).

To obtain performance estimates of each combination of classifier and input feature, a 10-fold cross-validation on the training data set was carried out when using grid or random search to optimize all hyperparameters (see Table 3; for details see Appendix A Table A1).

Table 3. Grid of hyperparameters to optimize the performance of the base classifiers acting on different textual and taxonomic features in 10-fold cross-validation.

Classifier	Hyperparameter	
	Feature: Text	
SVM kNN LogReg ANN	$nu \in \{0.1, 0.2, \ldots, 0.9\}$ $k \in \{1, 5, 10, 15, \ldots, 50\}$ $c \in \{10^{-6}, 10^{-5}, \ldots, 10^{5}\}$ $\alpha \in \{10^{-4}, 10^{-3}, \ldots, 10^{2}\}$ $hl \in \left\{ \begin{array}{l} (50,-),(100,-),(200,-), \\ (50,12),(100,12),(200,12) \end{array} \right\}$	$n-gram \in \{(1,1), \ldots, (3,3)\}$ $norm \in \{l2, l1, none\}$ $smooth\_idf \in \{true, false\}$ $sublin\_tf \in \{true, false\}$ $use\_idf \in \{true, false\}$
	Feature: Taxonomy	
SVM kNN LogReg ANN	$nu \in \{0.1, 0.2, \ldots, 0.9\}$ $k \in \{1, 5, 10, 15, \ldots, 50\}$ $c \in \{10^{-6}, 10^{-5}, \ldots, 10^{5}\}$ $\alpha \in \{10^{-4}, 10^{-3}, \ldots, 10^{2}\}$ $hl \in \left\{ \begin{array}{l} (50,-),(100,-),(200,-), \\ (50,12),(100,12),(200,12) \end{array} \right\}$	$n\_prototype \in \{10, 20, 50, 100, 200\}$ $prototype\_selection \in \left\{ \begin{array}{l} classwise\ random, \\ classwise\ spanning \end{array} \right\}$

For hyperparameter tuning and all further evaluations conducted on the externally validated test data set, we employ the overall micro-averaged F1 score [7], representing a balanced ratio of recall and precision. The ranking of the classification performances (F1) in training and testing provided an ordered list according to the summed-up ranks. The top and bottom ranked classifier from this list were selected for later combination following the notion, that in some cases even less well performing base learners might contribute substantially to the overall ensemble outcome [8].

3.3.2. Fusion Methods

A combination of classifiers $C_1, \ldots, C_m$ can be defined as:

$$C(x) = \mathcal{F}(C_1(x), \ldots C_m(x)) \tag{2}$$

Accordingly, the fusion method $\mathcal{F}$ systematically combines the predictions of the used base classifiers $C_1, \ldots, C_m$ processing an input object x. Two approaches to obtain a fusion method $\mathcal{F}$ are used.

Fixed combination rules combine the class prediction output of base classifiers in a predefined way by utilizing simple arithmetic operations (sum, product) or basic set operations (minimum, maximum).

Considering the classification output for input object x and class ω_i of base classifier C_j as $c_{j,i}(x)$ the sum rule is given by:

$$\mu_i(x) = \frac{1}{n} \sum_{j=1}^{n} c_{j,i}(x) \quad (3)$$

The product, minimum, and maximum fixed combination rules are constructed likewise [19].

Stacking interprets the combination of base classifiers' output as further classification problem creating a meta-classifier. The base classifiers' output is fed into a classifier for objects with known class assignment. In comparison to the fixed combination rules, this requires an additional training phase including hyperparameter tuning (for details see Appendix A Table A2) for the used meta-classifier while providing more flexibility to adapt to the learning data.

3.3.3. Experimental Design

The final evaluation consists of a variety of differently composed ensembles, which always consist of two base learners each. These ensembles differ in three factors, namely the feature used, the performance of the base learners, and the applied fusion method (see Table 4). Overall, 64 different ensembles were tested and the corresponding F1 score was computed.

Table 4. Factors of the experimental design to build the varying classifier ensembles.

Factor	Levels
Feature Source	Text; Taxonomy (IPC or CPC)
Base Performance Ranking	Top; Bottom
Fusion Method	Stacking (SVM, kNN, LogReg, ANN); Fixed Combination Rules (sum, product, min, max)

4. Results and Discussion

4.1. Basic Evaluation

The initial basic evaluation on the test data highlights the mutual dependency between the classification algorithm and the type of feature (see Figure 4). SVM and kNN perform best on text (69.1%) while ANN reaches nearly the same top score on CPC (68.1%) as Logistic Regression (LogReg) on IPC taxonomy. In contrast, the weaker base classifiers only achieve F1 values between 53.2% (CPC-kNN), 54.3% (IPC-kNN) and 60.6% for LogReg on text.

Considering the results of prior research [12], it was to be expected that SVM would again prove to be a powerful tool for text classification in the present task. The almost equal quality of classification achieved with both patent taxonomies using ANN or LogReg is all the more remarkable, because it is only about one percentage point below the best value of SVM on text. Different from the majority of past approaches that use the taxonomies as target of automated categorization [11,14], our results place both IPC and CPC close to the text of patent documents to be used as a valuable source of information suitable for machine learning approaches in multiclass environments.

Since medical technology is an international domain, its technology management is forced to analyze patents worldwide originally written in many different languages. However, in many countries, such as Germany, patents are not translated into English by default. Thus, only 51.6% of all records in the international patent database provided by the European Patent Office (PATSTAT 2019 autumn version [32]) contain an English title and abstract. This is a drawback for international technology management, as automated patent categorization based on textual features usually depends on the common language of the data.

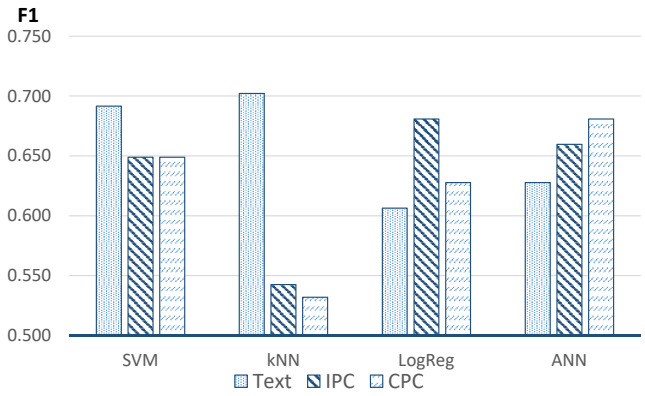

Figure 4. Performance of four different base classifiers using text or taxonomic (IPC, CPC) features comparing the overall F1 scores. Biomedical patents of the test data set were classified. IPC: International Patent Classification.

In contrast, CPC or IPC codes are language independent and very frequently assigned to the patent documents. In the PATSTAT database 85.2% of documents are assigned with ICP or CPC codes. Therefore, the IPC/CPC taxonomies could potentially bridge the language gap by even replacing text features in machine learning classification systems of international patents. We have explored this prospect in a preliminary analysis, the results of which are reported in Section 4.4, Outlook.

4.2. Ensemble Evaluation

The fusion of base classifiers reveals in most cases (60/64) an improvement of classification performance on the test data compared to the best base classifier of each ensemble (see Table 5).

Table 5. Resulting performance (F1) of the ensemble classification displayed as pivot table of different base classifiers combinations using stacking or fixed combinations rules as fusion method. The combination of the base learners in pairs of one text and one taxonomic (either IPC or CPC) classifier is additionally varied according to the Base Performance Ranking (BPR: top or bottom rank). For each of those conditions four different stacking classifiers as well four fixed-combinations rules are applied (see Table 4). Peak values are printed in bold.

FEATURE		**TEXT**			**TEXT**		
		BPRbottom		**BPRtop**	**BPRbottom**		**BPRtop**
Base Classifier		LogReg		SVM	LogReg		SVM
F1		0.606		0.691	0.606		0.691
Fusion method			**Stacking**			**Fixed Combination**	
IPC			**Fusion**			**Fusion**	
BPRbottom		0.691	SVM	0.702	0.681	sum	0.702
kNN	0.543	0.670	kNN	0.681	0.660	product	0.713
		0.660	LogReg	0.702	0.649	min	0.681
		0.691	ANN	0.745	0.681	max	0.702
BPRtop		0.713	SVM	0.755	0.734	sum	0.745
LogReg	0.681	0.745	kNN	0.745	0.734	product	0.745
		0.755	LogReg	0.755	0.702	min	0.713
		0.766	**ANN**	**0.787**	0.723	max	0.723
CPC							
BPRbottom		0.638	SVM	0.723	0.681	sum	0.702
kNN	0.532	0.691	kNN	0.713	0.660	product	0.681
		0.670	LogReg	0.745	0.660	min	0.681
		0.670	ANN	0.745	0.638	max	0.691
BPRtop		0.681	SVM	0.766	0.766	**sum**	**0.777**
ANN	0.681	0.766	kNN	0.745	0.734	**product**	**0.777**
		0.745	LogReg	0.777	0.713	min	0.745
		0.702	**ANN**	**0.787**	0.734	max	0.755

The improvements reach a maximum of 9.6%, which in two cases leads to the peak value of the entire performance matrix of F1 = 78.7%, namely by combining SVM (Text) with ANN (IPC) or ANN (CPC). At this stage, two first conclusions can be drawn: (1) in our approach, ensemble classification using text and taxonomic features in general proves to be a very effective method to enhance the overall classification performance; (2) our achieved maximum value certainly holds up to the comparison to multi-classifier fusion with F1 = 77.6% [13] before adding active learning components or network-based classification with F1 = 76.2% [7]. The latter approach only reaches F1 = 86.4% by hybridization of four different patent classification approaches. The impressive performance is counterbalanced by a high complexity in using a wide variety of methods and distinct data sources. In contrast, for our model the IPC/CPC codes and titles/abstracts of the patents are readily available from the same structured database [32]. Except DSE, the deployed feature transformations, the base classifiers, and fusion methods are all well-established, which further strengthen the applicability and efficiency of the presented pipeline. Finally, the fusion by the fixed combination rules sum and product achieved remarkable top scores of F1 = 77.7%, only one mark behind the overall peak using stacking, without the burden of training and hyperparameter tuning.

4.3. Boosting

To analyze the impact of the performance rank of base learners on the final ensemble outcome, the F1-results are averaged over this basic condition. They show a clear trend (see Table 6): The best performances are achieved by combining two top ranked learners in both fusion methods, stacking and fixed combination rules. Consistently, combining only bottom ranked classifiers places last in the ranking of the final outcome.

Table 6. Averaged F1 scores for the performance condition (Base Performance Ranking, BPR with top or bottom rank) of the four different stacking techniques and the four fixed combinations rules for all ensemble conditions of combining two base learners. Peak values are printed in bold.

FEATURE		**TEXT**		**TEXT**	
		BPRbottom	**BPRtop**	**BPRbottom**	**BPRtop**
Fusion method		**Stacking**		**Fixed Combination**	
		avg-F1			
IPC	**BPRbottom**	0.678	0.708	0,668	0.700
	BPRtop	0.745	**0.761**	0.723	**0.732**
CPC	**BPRbottom**	0.667	0.732	0.660	0.689
	BPRtop	0.724	**0.769**	0.737	**0.764**

Hence, we conclude, that in our case comparably weak base classifiers are not capable to strengthen the ensemble categorization. This somehow contradicts the fundamental notion which underlies the well-established boosting approaches, that have been just recently applied very successfully, e.g., in stream data classification and concept shift problems. However, the specific characteristic of boosting is that ‚weak learners' are defined as performing only slightly better than random categorization. For example, Lee and colleagues [37] applying boosting to predict the transfer potential of patents finally reach a top F1 score of 0.589. Compared to our approach, even the bottom ranked base learners perform at the same level (F1 = 0.532) and can consequently not be seen as truly 'weak' classifiers. In general, it has been proven that AdaBoost is not successful with strongly performing base learners [38].

4.4. Outlook

As discussed with the basic evaluation above, the best taxonomic base learners are performing nearly as well as the still leading text-based classifier SVM (see Table 5). This raises the question whether a combination of just strong IPC and CPC-based classifiers, omitting all text features, might boost the overall performance, perhaps even beyond the scores of

the best text-based learners. Thus, a corresponding preliminary experiment has been conducted: the base learners LogReg (IPC) and ANN (CPC) are combined by fixed combination rules as well as using stacking (see Figure 5).

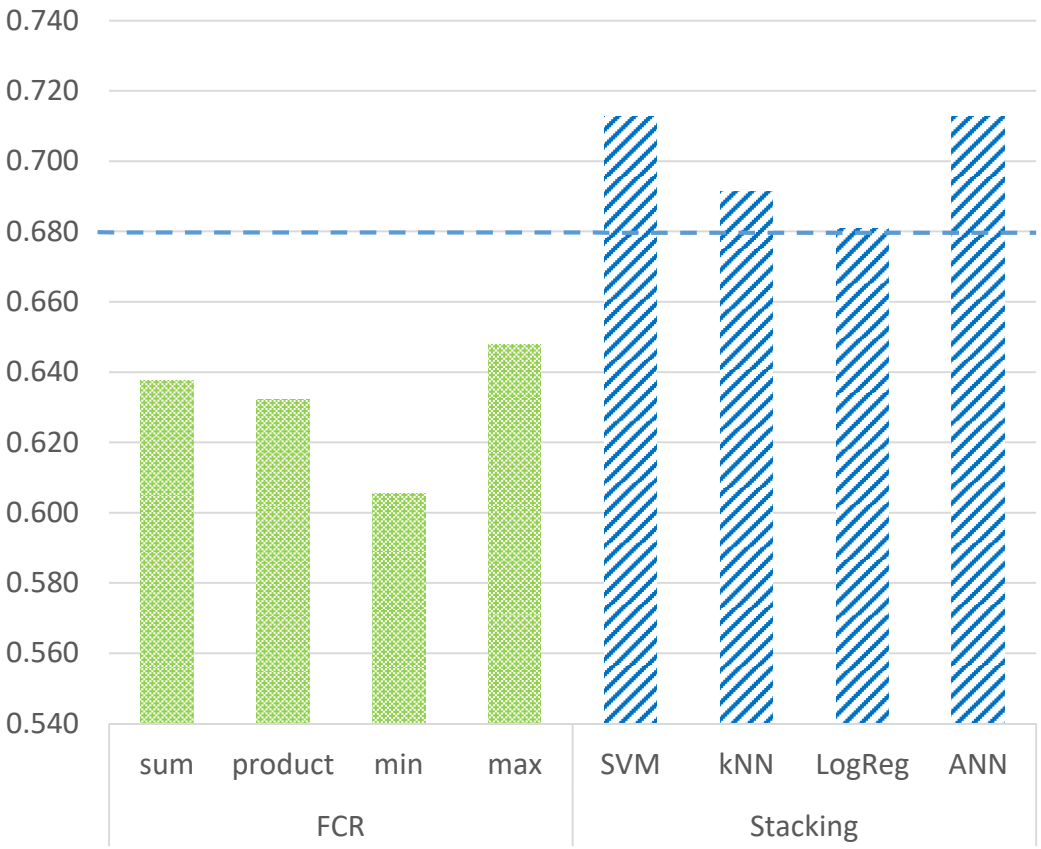

Figure 5. Performance of language independent ensemble classification (F1) by combining the two top ranked IPC and CPC base learners with different fusion methods: four different stacking classifiers as well four fixed-combinations rules (FCR) are applied. The dotted line marks the top performance value of the base learners.

Whereas the fusion with FCR provides no improvement, the stacking with both ANN and SVM enhances the overall performance up to an F1 score of 71.2% on the test data. This outperforms the so far best basic performance of a text-based SVM and is completely independent from the language of the written patent information. Considering the PATSTAT patent database, now more than 22 million additional documents, lacking English titles/abstracts, could be included into a technology analysis based on automated categorization. The potential seems to be very large, but beforehand further studies have to evaluate the scalability of our preliminary results to further domains and larger sample sizes.

Furthermore, modern deep learning approaches such as BERT or ELMo [39,40], with extensive pre-trained language models including multilingual and domain-specific variants (e.g., BioBERT [41]), show a great potential in semantic tasks and might boost our patent classification also in terms of cross-lingual patent data. However, it remains to be clarified whether the high effort of such complex approaches is also worthwhile in terms of applicability and performance. The results and data of our work might serve as basis for an elaborate evaluation as part of future work.

4.5. Limitations

In this paper, for the first time—to the best of our knowledge—IPC and CPC taxonomies are applied as graph-based information resources for automated patent classifica-

tion. Despite the success, some limitations remain. Although we conducted an extensive hyperparameter tuning, the cost of 1 for each operation to calculate tree edit distance stayed untuned. Altering the costs in preliminary experiments modified the overall performance and should therefore be investigated in future work. Likewise, using solely the CPC/IPC's string representation to parse the corresponding tree will not extract all the available information. Thus, with a method to display the full subgroup relationship more information could be inserted to the CPC code tree.

5. Conclusions

We were able to show that the official patent taxonomies, IPC and CPC, contribute substantially to the performance of automated patent categorization into user-defined classes when the graph is transformed into a real-valued vector space by Dissimilarity Space Embedding. The multiclass classification with taxonomic base classifiers achieves F1 values close to the top text classifier. The fusion of the best performing taxonomic and textual base classifiers results in an overall increase in performance by +9.6% to a final F1 score of 78.7%.

This does not only unlock the potential of hierarchical patent taxonomies as valuable part of an ensemble to enhance automated patent categorization. In addition, ICP and CPC are language-independent, which makes the difference: the solely taxonomic ensemble with stacking performs even better than the best text-based learners. This opens the access to millions of additional patent documents enabling new possibilities for the management of biomedical technologies.

Overall, the deployed methods are well established in research and all used data are accessible from one integrated source. Even the novel implementation of the taxonomic CPC and IPC graph is built upon a well-descripted procedure (DSE) to transform the information into a real-valued vector. Consequently, this increases the applicability of the whole approach.

Our novel approach could also make an important contribution to the digitization of the health care system. For the annotation of health data, medical category systems such as the International Classification of Diseases or SNOMED CT are used. Accordingly, this graph-based information could now contribute to future solutions using the presented approach, thus advancing AI implementations in decision support for diagnosis and therapy.

Author Contributions: Design of the approach, software, investigation and draft manuscript preparation, K.F.; data curation, software testing, formal analysis, M.B.; visualization, S.G.; validation, resources, supervision, R.F. All authors have cooperated in conceptualization as well as in editing and revising the paper. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by Klaus Tschira Foundation gGmbH, Heidelberg, Germany.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Restrictions apply to the availability of these data. Data were obtained from European Patent Office (EPO) and are available at https://www.epo.org/searching-for-patents/business/patstat.html with the permission of EPO.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Table A1. Results of hyperparameter tuning of the base classifiers: feed-forward Artificial Neural Network (ANN), k-Nearest-Neighbor (kNN), Logistic Regression (LogReg), and Support Vector Machine (SVM). The F1 scores for testing and 10-fold cross-validation (CV) were ranked (values in parentheses: Base Performance Ranking, BPR) for each feature. The summed BPR are used to select top (**) and bottom (*) ranked classifiers to build ensemble pairs for final evaluation.

Classifier	Optimized Hyperparameters	CV (BPR)	Test (BPR)
	Feature: Text		
ANN	hidden_layer_sizes = (200,12), alpha = 1, max_iter = 10,000, *n*-gram = (1,1), norm = none, smooth_idf = true, sublinear_tf = true, use_idf = true	84.1% (2)	62.8% (3)
kNN	k = 40, *n*-gram = (1,1), norm = L2, smooth_idf = false, sublinear_tf = true, use_idf = true	79.3% (4)	70.2% (1)
LogReg *	c = 10, solver = saga, multi-class = multinomial, max_iter = 10,000, *n*-gram = (1,1), norm = none, smooth_idf = false, sublinear_tf = true, use_idf = true	84.6% (1)	60.6% (4)
SVM **	nu = 0.6, kernel = RBF, *n*-gram = (1,1), norm = none, smooth_idf = false, sublinear_tf = true, use_idf = true	82.9% (3)	69.1% (2)
	Feature: IPC Taxonomy		
ANN	hidden_layer_sizes = (100,12), alpha = 1, max_iter = 10,000, *n*_prototype = 200, prototype_selection = classwise_random	79.8% (1)	66.0% (2)
kNN *	k = 5, *n*_prototype = 100, prototype_selection = classwise_spanning	69.3% (4)	54.3% (4)
LogReg **	c = 1, solver = saga, multi-class = multinomial, max_iter = 10,000, *n*_prototype = 200, prototype_selection = classwise_random	80.1% (1)	68.1% (1)
SVM	nu = 0.2, kernel = RBF, *n*_prototype = 200, prototype_selection = classwise_random	77.1% (3)	64.9% (3)
	Feature: CPC Taxonomy		
ANN **	hidden_layer_sizes = (50,12), alpha = 1, max_iter = 10,000, *n*_prototype = 200, prototype_selection = classwise_spanning	74.4% (2)	68.1% (1)
kNN *	k = 5, *n*_prototype = 100, prototype_selection = classwise_spanning	60.4% (4)	53.2% (4)
LogReg	c = 1, solver = saga, multi-class = multinomial, max_iter = 10,000, *n*_prototype = 100, prototype_selection = classwise_random	75.1% (1)	62.8% (3)
SVM	nu = 0.2, kernel = RBF, *n*_prototype = 100, prototype_selection = classwise_spanning	72.0% (3)	64.9% (2)

Table A2. Results of hyperparameter tuning of ensembles using stacking with four different fusion classifiers: feed-forward Artificial Neural Network (ANN), k-Nearest-Neighbor (kNN), Logistic Regression (LogReg), and Support Vector Machine (SVM). The base learners in pairs, one text and one taxonomic (either IPC or CPC) classifier, is varied according to the Base Performance Ranking (BPR: top or bottom rank). The grid search with 10-fold cross-validation (CV) was performed to maximize F1. 'hl' stands for 'hidden_layer_sizes'.

FEATURE		**TEXT**				
		BPRbottom			**BPRtop**	
Base Classifier		LogReg			SVM	
	Fusion	**F1**	**Hyperparameter**	**F1**	**Hyperparameter**
TAXONOMIES					
IPC kNN	SVM	0.846	nu = 0.85	0.841	nu = 0.9
	kNN	0.843	k = 50	0.841	k = 20
	LogReg	0.846	c = 1.0	0.846	c = 0.01
	ANN	0.847	alpha = 1, hl = (100,12)	0.843	alpha = 10, hl = (10,-)
BPRbottom					
CPC kNN	SVM	0.837	nu = 0.85	0.831	nu = 0.7
	kNN	0.843	k = 25	0.830	k = 50
	LogReg	0.838	c = 0.01	0.828	c = 0.1
	ANN	0.841	alpha = 10, hl = (50,-)	0.831	alpha = 0.1, hl = (10,12)
IPC LogReg	SVM	0.859	nu = 0.9	0.845	nu = 0.25
	kNN	0.859	k = 50	0.854	k = 30
	LogReg	0.864	c = 0.001	0.853	c = 0.01
	ANN	0.863	alpha = 10, hl = (10,-)	0.853	alpha = 10, hl = (100,12)
BPRtop					
CPC ANN	SVM	0.844	nu = 0.55	0.852	nu = 0.25
	kNN	0.846	k = 15	0.843	k = 25
	LogReg	0.849	c = 0.1	0.842	c = 1.0
	ANN	0.850	alpha = 0.1, hl = (10,-)	0.844	alpha = 10, hl = (10,-)

References

1. Kreuchauff, F.; Korzinov, V. A patent search strategy based on machine learning for the emerging field of service robotics. *Scientometrics* **2017**, *111*, 743–772. [CrossRef]
2. Jaffe, A.B.; De Rassenfosse, G. Patent citation data in social science research: Overview and best practices. *J. Assoc. Inf. Sci. Technol.* **2017**, *68*, 1360–1374. [CrossRef]
3. Leydesdorff, L.; Kogler, D.F.; Yan, B. Mapping patent classifications: Portfolio and statistical analysis, and the comparison of strengths and weaknesses. *Scientometrics* **2017**, *112*, 1573–1591. [CrossRef] [PubMed]
4. Schmoch, U. Concept of a Technology Classification for Country Comparisons–Final Report to the World Intellectual Property Organisation (WIPO). Available online: https://www.wipo.int/export/sites/www/ipstats/en/statistics/patents/pdf/wipo_ipc_technology.pdf (accessed on 12 February 2020).
5. Wolpert, D.H. *The Supervised Learning No-Free-Lunch Theorems*; Springer: Berlin/Heidelberg, Germany, 2002; pp. 25–42.
6. Sebastiani, F. Machine learning in automated text categorization. *ACM Comput. Surv.* **2002**, *34*, 1–47. [CrossRef]
7. Liu, D.-R.; Shih, M.-J. Hybrid-patent classification based on patent-network analysis. *J. Am. Soc. Inf. Sci. Technol.* **2010**, *62*, 246–256. [CrossRef]
8. Duin, R.P.W. The Combining Classifier: To Train or Not to Train? In *16th International Conference on Pattern Recognition (ICPR 2002), Proceedings of the 16th International Conference on Pattern Recognition, Quebec, QC, Canada, 11–15 August 2002*; IEEE Imprint: Los Alamitos, CA, USA, 2002; pp. 765–770. ISBN 0-7695-1695-X.
9. Riesen, K.; Bunke, H. Graph Classification Based on Vector Space Embedding. *Int. J. Pattern Recognit. Artif. Intell.* **2009**, *23*, 1053–1081. [CrossRef]
10. Trappey, A.; Hsu, F.-C.; Trappey, C.V.; Lin, C.-I. Development of a patent document classification and search platform using a back-propagation network. *Expert Syst. Appl.* **2006**, *31*, 755–765. [CrossRef]
11. Fall, C.J.; Törcsvári, A.; Benzineb, K.; Karetka, G. Automated categorization in the international patent classification. *ACM SIGIR Forum* **2003**, *37*, 10–25. [CrossRef]
12. Anne, C.; Mishra, A.; Hoque, T.; Tu, S. Multiclass patent document classification. *Artif. Intell. Res.* **2017**, *7*, 1. [CrossRef]
13. Zhang, X. Interactive patent classification based on multi-classifier fusion and active learning. *Neurocomputing* **2014**, *127*, 200–205. [CrossRef]
14. Tran, T.; Kavuluru, R. Supervised Approaches to Assign Cooperative Patent Classification (CPC) Codes to Patents. In *MIKE*; Springer: Berlin/Heidelberg, Germany, 2017; Volume 10682, pp. 22–34.
15. Lee, J.-S.; Hsiang, J. Patent classification by fine-tuning BERT language model. *World Pat. Inf.* **2020**, *61*, 101965. [CrossRef]

16. Woźniak, M.; Graña, M.; Corchado, E. A survey of multiple classifier systems as hybrid systems. *Inf. Fusion* **2014**, *16*, 3–17. [CrossRef]
17. Tulyakov, S.; Jaeger, S.; Govindaraju, V.; Doermann, D. Review of Classifier Combination Methods. In *Machine Learning in Document Analysis and Recognition*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 361–386.
18. Rokach, L. Ensemble-based classifiers. *Artif. Intell. Rev.* **2010**, *33*, 1–39. [CrossRef]
19. Kuncheva, L.I. *Combining Pattern Classifiers*; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2014; ISBN 9781118914564.
20. Kuncheva, L.I.; Whitaker, C.J. Measures of Diversity in Classifier Ensembles and Their Relationship with the Ensemble Accuracy. *Mach. Learn.* **2003**, *51*, 181–207. [CrossRef]
21. Kuncheva, L. A theoretical study on six classifier fusion strategies. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 281–286. [CrossRef]
22. Džeroski, S.; Ženko, B. Is Combining Classifiers with Stacking Better than Selecting the Best One? *Mach. Learn.* **2004**, *54*, 255–273. [CrossRef]
23. Freund, Y.; Schapire, R.E. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *J. Comput. Syst. Sci.* **1997**, *55*, 119–139. [CrossRef]
24. Li, X.; Wang, L.; Sung, E. AdaBoost with SVM-based component classifiers. *Eng. Appl. Artif. Intell.* **2008**, *21*, 785–795. [CrossRef]
25. Sun, J.; Li, H.; Fujita, H.; Fu, B.; Ai, W. Class-imbalanced dynamic financial distress prediction based on Adaboost-SVM ensemble combined with SMOTE and time weighting. *Inf. Fusion* **2020**, *54*, 128–144. [CrossRef]
26. Santos, S.G.T.D.C.; De Barros, R.S.M. Online AdaBoost-based methods for multiclass problems. *Artif. Intell. Rev.* **2020**, *53*, 1293–1322. [CrossRef]
27. Foggia, P.; Percannella, G.; Vento, M. Graph Matching and Learning in Pattern Recognition in the Last 10 Years. *Int. J. Pattern Recognit. Artif. Intell.* **2014**, *28*, 1450001. [CrossRef]
28. Wilson, R.C.; Hancock, E.R.; Luo, B. Pattern vectors from algebraic graph theory. *IEEE Trans. Pattern Anal. Mach. Intell.* **2005**, *27*, 1112–1124. [CrossRef] [PubMed]
29. Riesen, K.; Ferrer, M.; Fischer, A. Building Classifier Ensembles Using Greedy Graph Edit Distance. In *Multiple Classifier Systems, Proceedings of the 12th International Workshop, MCS 2015, Günzburg, Germany, 29 June–1 July 2015*; Schwenker, F., Roli, F., Kittler, J., Eds.; Springer International Publishing: Cham, Switzerland, 2015; pp. 125–134. ISBN 978-3-319-20247-1.
30. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
31. Bukowski, M.; Geisler, S.; Schmitz-Rode, T.; Farkas, R. Feasibility of activity-based expert profiling using text mining of scientific publications and patents. *Scientometrics* **2020**, *123*, 579–620. [CrossRef]
32. European Patent Office. PATSTAT. Available online: https://www.epo.org/searching-for-patents/business/patstat.html (accessed on 12 February 2020).
33. Bille, P. A survey on tree edit distance and related problems. *Theor. Comput. Sci.* **2005**, *337*, 217–239. [CrossRef]
34. European Patent Office and United States Patent and Trademark Office. CPC Scheme and Definitions. Available online: https://www.cooperativepatentclassification.org/cpcSchemeAndDefinitions (accessed on 12 February 2020).
35. Zeng, Z.; Tung, A.K.H.; Wang, J.; Feng, J.; Zhou, L. Comparing stars: On Approximating Graph Edit Distance. *Proc. VLDB Endow.* **2009**, *2*, 25–36. [CrossRef]
36. Zhang, K.; Shasha, D. Simple Fast Algorithms for the Editing Distance between Trees and Related Problems. *SIAM J. Comput.* **1989**, *18*, 1245–1262. [CrossRef]
37. Lee, J.; Kang, J.-H.; Jun, S.; Lim, H.; Jang, D.-S.; Park, S. Ensemble Modeling for Sustainable Technology Transfer. *Sustainability* **2018**, *10*, 2278. [CrossRef]
38. Wickramaratna, J.; Holden, S.; Buxton, B. Performance Degradation in Boosting. In *Multiple Classifier Systems, Proceedings of the Second International Workshop, MCS 2001, Cambridge, UK, 2–4 July 2001*; Kittler, J., Ed.; Springer: Berlin/Heidelberg, Germany, 2001; pp. 11–21, ISBN 978-3-540-42284-6.
39. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. 2018. Available online: http://arxiv.org/pdf/1810.04805v2 (accessed on 7 January 2021).
40. Peters, M.E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; Zettlemoyer, L. Deep contextualized word representa-tions. 2018. Available online: http://arxiv.org/pdf/1802.05365v2 (accessed on 7 January 2021).
41. Lee, J.; Yoon, W.; Kim, S.; Kim, D.; Kim, S.; So, C.H.; Kang, J. BioBERT: A pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics* **2019**, *36*, 1234–1240. [CrossRef]