Kreuchauff, Florian; Korzinov, Vladimir

Working Paper

A patent search strategy based on machine learning for the emerging field of service robotics

KIT Working Paper Series in Economics, No. 71

Provided in Cooperation with:
Karlsruhe Institute of Technology (KIT), Institute of Economics (ECON)

Suggested Citation: Kreuchauff, Florian; Korzinov, Vladimir (2015) : A patent search strategy based on machine learning for the emerging field of service robotics, KIT Working Paper Series in Economics, No. 71, Karlsruher Institut für Technologie (KIT), Institut für Volkswirtschaftslehre (ECON), Karlsruhe,
https://doi.org/10.5445/IR/1000049790

This Version is available at:
https://hdl.handle.net/10419/120880

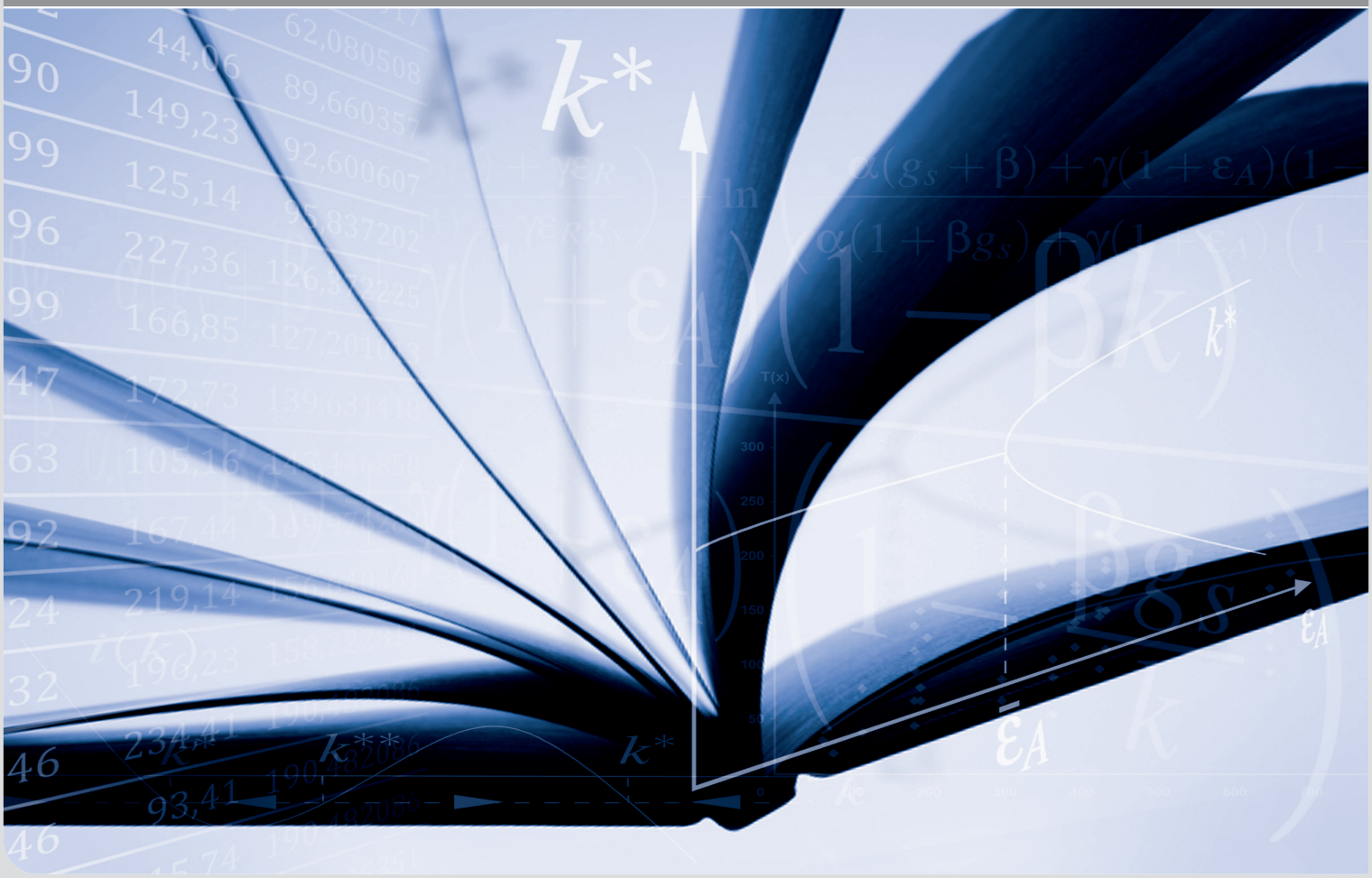
Karlsruher Institut für Technologie

A patent search strategy based on machine learning for the emerging field of service robotics

by Florian Kreuchauff and Vladimir Korzinov

No. 71 | AUGUST 2015

WORKING PAPER SERIES IN ECONOMICS



KIT – University of the State of Baden-Wuerttemberg and
National Laboratory of the Helmholtz Association

econpapers.wiwi.kit.edu

Impressum

Karlsruher Institut für Technologie (KIT)
Fakultät für Wirtschaftswissenschaften
Institut für Volkswirtschaftslehre (ECON)

Schlossbezirk 12
76131 Karlsruhe

KIT – Universität des Landes Baden-Württemberg und
nationales Forschungszentrum in der Helmholtz-Gemeinschaft

Working Paper Series in Economics
No. 71, August 2015

ISSN 2190-9806

econpapers.wiwi.kit.edu

A Patent Search Strategy based on Machine Learning for the Emerging Field of Service Robotics

Florian Kreuchauff
Karlsruhe Institute of Technology (KIT) **

Vladimir Korzinov
Karlsruhe Institute of Technology §

August 31, 2015

Abstract

Emerging technologies are in the core focus of supra-national innovation policies. These strongly rely on credible data bases for being effective and efficient. However, since emerging technologies are not yet part of any official industry, patent or trademark classification systems, delineating boundaries to measure their early development stage is a nontrivial task. This paper is aimed to present a methodology to automatically classify patents as concerning service robots. We introduce a synergy of a traditional technology identification process, namely keyword extraction and verification by an expert community, with a machine learning algorithm. The result is a novel possibility to allocate patents which (1) reduces expert bias regarding vested interests on lexical query methods, (2) avoids problems with citational approaches, and (3) facilitates evolutionary changes. Based upon a small core set of worldwide service robotics patent applications we derive apt n-gram frequency vectors and train a support vector machine (SVM), relying only on titles, abstracts and IPC categorization of each document. Altering the utilized Kernel functions and respective parameters we reach a recall level of 83% and precision level of 85%.

Key words: Service Robotics, Search Strategy, Patent Query, Data Mining, Machine Learning, Support Vector Machine

JEL-codes: C02; C18; C45

** florian.kreuchauff@kit.edu
§ vladimir.korzinov@kit.edu

Acknowledgments

We are thankful to the High Performance Humanoid Technologies (H2T) group from the Institute for Anthropomatics and Robotics at Karlsruhe Institute of Technology in Germany, in particular to Prof. Dr. Tamim Asfour and Prof. Dr. Gabriel Lopes from Delft Center for Systems and Control / Robotics Institute at TU Delft in the Netherlands for their support and advices. Moreover, we wish to thank the participants of the 15th EBES conference in Lisbon as well as the participants of the 6th annual S.NET meeting in Karlsruhe for their valuable comments and suggestions that have led to the improvement of this article. This work is supported by the project "Value Creation & Innovation Processes in and beyond Technology" of the Karlsruhe School of Services.

1 Introduction

Innovation policies that address promising emerging technologies serve to reach macroeconomic objectives such as promoting sustainable growth and prosperity. They are legitimated due to the various uncertainties associated with new technological fields that result from coordination problems in complex innovation chains with scale economies, multilateral dependencies, and externalities. In order to develop effective policy measures, one has to carefully recognize emergence patterns and assess possible downstream effects. This is a demanding task since these patterns vary across technologies, time, scale, and regional and institutional environments. It is important that policy advices rely on credible data sources that aptly display early research and innovation results at the very beginning of value creation. However, as long as a new technology has not yet been specified within official statistical schemes, the identification of delineating boundaries in respective data bases is a nontrivial problem.

Service robotics (hereafter SR) is a current example of an emerging technology. The International Federation of Robotics (IFR) has been working on a service robot definition and classification scheme since 1995. A preliminary definition states that a service robot is a robot that performs useful tasks for humans or equipment excluding industrial automation applications. Industrial automation applications include, but are not limited to, manufacturing, inspection, packaging, and assembly (compare `www.ifr.org` and ISO 8373:2012). Service robots can be further subdivided into those for non-commercial personal use like domestic servant robots or automated wheelchairs, and those for professional commercial services, for which they are usually run by trained operators like fire-fighting or surgery

systems in hospitals. Hence, SR contribute to both traditional and a variety of new types of services.[1]

As a result of the arising multiplicity, the technology field so far is not clearly confined in databases and thus neither part of any existing official industry, patent or trademark classification system nor of any concordances not to mention national account systems. Having said that, distinguishing SR from industrial robotics (hereafter IR) is hardly possible. This so far has impeded a comprehensive assessment of the economic impacts of SR diffusion, especially with respect to the magnitude, timing and geographical localization.

With our work we make SR tractable by developing a search strategy to identify it within patent databases. Moreover, we model the approach not to be limited to patents but to be applicable for scientific publications as well. In addition, the general methodology is not even confined to the field of robotics, but could be applied to any similar identification problem. Differentiating from classical lexical and citational approaches used by other scholars our approach introduces a machine learning algorithm that is utilized as a classifier. Being trained on some sample data this classifier acts as an 'expert'. The machine is able to decide whether a patent belongs to the category of service robotics or not – with a certain degree of precision. Since there are several approaches in the scientific literature which deal with analogous problems of technology detection and classification, we hereby set out to (1) limit expert bias regarding vested interests on lexical query methods (with respect to term inclusion and exclusion), (2) avoid problems with citational approaches such as the lack of portability, and (3) facilitate evolutionary changes.

The following sections are organized as follows: First, we give an overview of previous technology identification approaches referring to examples of similar emerging fields that lacked classification schemes in its infant phase. Second, we present our step-by-step methodology for identifying developments in an emerging field characterized only by its early applications. It successively describes the use of patents as apt data source, the retrieval of a structured core dataset, and the use of an automated machine learning algorithm, namely a support vector machine (hereafter SVM). Finally, we present results of our pioneering approach and conclude with future scope for improvement.

[1] Beyond its potential productivity effects SR is believed to induce visible changes in employment structures (Autor et al. 2003, Frey and Osborne 2013, Graetz and Michaels 2015). Its potential to change organization processes in firms as well as everyday life of people is already visible in the diffusion of semi-autonomous physical systems out of industrial fabrication and into service economies.

2 Literature Review

There is no widely agreed-upon definition of emerging technologies (Halaweh 2013). The initial lack of common knowledge, standards, and specifications entails uncertainties along various dimensions (Stahl 2011). Future costs and benefits, relevant actors, adoption behaviour, and potential socio-economic implications such as creative destruction are highly unclear (Srinivasan 2008). Therefore, scientific studies have been using bibliometrics to monitor trends for a variety of domains and assess the nature of emerging technologies already within scientific research and early development.

No matter what the paramount aim, all analyses greatly rely on well-founded data acquisition, which first and foremost identifies the technology under consideration. With ongoing technological advancements as well as computational power more and more elaborated strategies have accrued. Most often, technology detection within patent or publication databases is predicated on either (1) lexical, (2) citationist, or mixed search strategies.[2]

For example, early conceptions of apt queries for nanotechnology proved to be difficult, as the first specific IPC-subclass B82B[3], which basically refers to nano-structures and their fabrication, was not introduced before the year 2000 and did not incorporate applications from former years (Noyons et al. 2003). In its infancy, it contained only estimated 10 percent of all relevant documents. Hence, the first scientific identification approach for nanoscience and technology relied instead on a lexical query developed in 2001 by the Fraunhofer Institute for Systems and Innovation Research (ISI) in Germany and the Centre for Science and Technology Studies (CWTS) at Leiden University in the Netherlands.

A lexical query (1) is a search for specified terms, which in the most simple case might consist of only one word (like 'nano*' for nanotechnologies) or a basic combination (like 'service robot*'). This primal string is applied to titles, abstracts, keywords or even the whole text body of examined documents. Some of these documents might prove to be relevant in the eyes of experts and thus offer additional terms starting an iterative process.[4] Considering emerging fields the number of terms within a search string that is developed

[2]With respect to scientific publications another common strategy is to identify core journals. All articles within those journals are then considered relevant. For patents though, this search strategy is obviously not feasible, which is why we do not deepen it further.

[3]Only in 2011 a second sub-class, B82Y, focusing on specific uses or applications of nano-structures was introduced for IPC and the Cooperative Patent Classification (CPC). Previously, related nano patent documents could only be identified if they were classified via the European Classification System (ECLA) with the specific sub-class Y01N.

[4]Such a search strategy is called evolutionary, if subsequent researchers may build upon existing query structures by progressively incorporating terms that better specify the technology and widen its scope (Mogoutov and Kahane 2007).

in such a lexical manner naturally grows rapidly. More and more scholars and practitioners become attracted by the field [5] adding alternatives and broadening interpretations in the course of time. Referring to nanotechnology as a striking example again, in order to keep track of the dynamically spreading nano-fields Porter et al. (2008) comprised a modular Boolean keyword search strategy with multiple-step inclusion and exclusion processes, which had to be subsequently enhanced and evolutionary revised (Arora et al. 2013). Identification problems are heightened by the fact that both authors of scientific publications as well as applicants of patents are interested in some rephrasing: The former, because they might benefit from a serendipity effect if their label establishes itself in the scientific community. And the latter because of encryption and legalese issues: Applicants may want to re-label critical terms, both to hide relevant documents and technical information from actual rivals and to build patent thickets of overlapping IPR which precludes potential competitors from commercializing new technology altogether.

A lexical query can be enriched (or fully substituted, if a core of documents is already verified) by adding documents and inherent terms identified via citational approaches (2), for instance by including new publications that are cited by at least two authors belonging to an initial database (Garfield 1967, Bassecoulard et al. 2007)[6] or, regarding patents, by including applications that refer prior art that has been part of the previously established core. In our example, Mogoutov and Kahane (2007) enriched an initial nanostring by a number of subfields, automatically identified and defined through the journal inter-citation network density displayed in the initial core dataset of nano-documents. Relevant keywords linked to each subfield were then tested for their specificity and relevance before being sequentially incorporated to build a final query.

The instance of nanotechnology illustrates well how much effort the development of an evolutionary query yields. Lately, private interests – rather than governmental or scientific research - have driven even more elaborated technology identification procedures: Companies that seek to monitor competitors or investigate latest research trends have started to rely on more cost-efficient processes in order to lower resulting expenditures. As a side effect, some encompassing literature on specialized text mining techniques has emerged, which goes beyond lexical and citation based procedures. To name just a few, Li et al. (2009) attempt to find significant rare keywords considering heterogeneous terms used

[5] For the instance of nanotechnology, to which we refer throughout, Arora et al. (2014) measure the growth in nano-prefixed terms in scholarly publications and find that the percentage of articles using a nano-prefixed term has increased from less than 10% in the early 1990s to almost 80% by 2010.

[6] This approach naturally harbours the risk of including generic articles of any scientific field that somehow happen to be cited in a technologically unrelated context. Bassecoulard et al. (2007) therefore incorporate a statistical relevance limit relying on the specificity of citations.

by assignees, attorneys, and inventors. Yoon and Park (2004) argue that citation analysis has some crucial drawbacks and propose a network-based analysis as alternative method, that groups patents according to their keyword distances. Lee (2008) uses co-word analyses regarding term association strength and provides indicators and visualization methods to measure latest research trends. Lee et al. (2009) transform patent documents into structured data to identify keyword vectors, which they boil down to principal components for a low-dimensional mapping. These facilitate the identification of areas with low patent density, which are interpreted as vacancies and thus chances for further technical exploitation. Erdi et al. (2013) use methods of citation and social network analysis, cluster generation, and trend analysis. Tseng et al. (2007) attempt to develop a holistic process for creating final patent maps for topic analyses and other tasks such as patent classification, organization, knowledge sharing and prior art searches. They describe a series of techniques including text segmentation, summary extraction, feature selection, term association, cluster generation, topic identification and information mapping. Engineering research itself shares some interest in following latest developments as well. For the field of robotics, Ruffaldi et al. (2010) is a good instance: They visualize trends in the domains of rehabilitation and surgical robotics identified via text mining.

3 Methodology

Following Mogoutov and Kahane (2007), the relative performance of different identification approaches may be compared via the respective degree of intervention of experts, their portability, their transparency regarding core features and respective impacts on final results, their replicability, their adaptability, meaning its ability to produce valid results while the technology in question keeps evolving, their updating capacity, and the extent and relevance of the data obtained. Certainly, no single best approach exists, since any method has its advantages and drawbacks according to these criteria. We will conclude on the relative performance of our approach at the end of this paper.

In line with the current text mining literature we propose a machine learning algorithm instead of a purely lexical, purely citationist or mixed query. Consequently, we first identify a small core patent dataset consisting of 228 patent applications and then let automated algorithms identify emerging technology boarders.

Patents as Data Source

As soon as a technology is sufficiently well specified, generically distinguishable, and ideally properly classified there are various techniques to map ongoing advancements. How-

ever, if such a delineation is not yet established and no broadly accepted consensus has been reached so far, economists most often rely on lexical, citation based, or mixed search strategies for prior identification purposes that help to trace related emerging fundamental and application knowledge in academic articles and patent documents.[7] As regards the technology under consideration, it is important to acknowledge that according to the IFR, the intended use, and as a consequence, the factual field of application determines the delimitation of SR from IR. Thus, patents are the data source of choice for an automated SR identification, since patentability requires an indication of the intended commercial implementation. Patents, despite all difficulties that arise in their use and interpretation, are widely accepted as indicator for innovative activity (Griliches 1990, Hall et al. 2005). Especially citation structures facilitate tracing knowledge flows (see, for instance, Jaffe et al. 1993, Thompson 2006, Fischer et al. 2009, Bresnahan 2010) and thus make technology development patterns visible. Hence, we started with a patent search strategy with a vision to extrapolate it to other lexical sources.

Building a structured core dataset that is suited for later application in machine learning requires the identification of a sufficiently large number of documents, that are validated as part of the technology and capture most of its hitherto variety of developments. This validation is granted by independent technological experts, who can either provide those documents themselves or may be given a predefined assortment to adjudicate on. The latter decreases a potential expert bias with respect to multifaceted preferences but might give rise to a negative influence of the researcher himself, who has to develop a search method for this primal assortment. Facing this trade-off, we decided to provide experts with a predefined core dataset.

Retrieval of a Core SR Patent Dataset

All unstructured patent text data as well as related document meta data were extracted from the 'EPO Worldwide Patent Statistical Database' (PATSTAT), version April 2013.[8] First, we extracted all patents that were either sorted in IPC class B25J[9] or contained a substring like 'robot*' in their respective title or abstract.[10] Hence, we established a

[7] Consequently, the adequate data sources for this identification process are the same that comprise the targets of subsequent analyses which might give cause for some criticism.

[8] This database encompasses raw data about 60 million patent applications and 30 million granted patents, utility models, PCT applications, etc. filed at more than 100 patent authorities worldwide.

[9] MANIPULATORS; CHAMBERS PROVIDED WITH MANIPULATION DEVICES. See `http://www.wipo.int/classifications/ipc/en/`

[10] According to the USPTO, most of the manipulators classified in B25J are industrial robots. See `http://www.uspto.gov/web/patents/classification/cpc/html/defB25J.html`.

set of documents describing robotic devices. Second, in order to identify a subset of potential SR patent documents that comprise most of the hitherto existing developments we created 11 sub-queries based mainly upon IFR application fields for service robots. These queries consisted both of IPC subclasses (mostly on 4-digit-level) and stemmed lexical terms, combined modularly in a Boolean structure.[11]

The second step provided us with 11 non-disjunct subsamples of potential SR patents. While other approaches regarding similar tasks of technology identification from there on further evaluate candidate terms by testing, assessing and adjusting terms and class codes to address weaknesses and follow emerging research trails manually (Porter et al. 2008), we did not alter the primal modular Boolean search. Instead, as indicated above, we left it to technological experts to verify the underlying categorization. Two independent academic expert groups with 15 scientists, affiliated with the

- High Performance Humanoid Technologies (H2T) from the Institute for Anthropomatics and Robotics at KIT, Germany, and the
- Delft Center for Systems and Control / Robotics Institute at TU Delft, Netherlands,

took on the task to decide which of the patents belonged to SR and which belonged complementarily to IR. The above experts were specialized in humanoid robotics, computer science, and mechanical engineering. Their experience in the field of robotics varied between 1 and 15 years. We provided them with 228 full body versions of potential SR patents from all over the world, extracted with the primal subsample queries. All patents listed in PATSTAT disclose at least English titles and abstracts. Thus, the judging scientists could always refer to these text parts as well as to all engineering drawings, no matter what the language of the remaining text body was.

For the application of automated machine learning approaches we then transformed the unstructured patent document text into structured data. This included several steps, namely (1) combining titles and abstracts in one body and splitting the resulting strings into single terms in normal lower cases, (2) removing stop words, (3) stemming, i.e. reducing inflected words to their stem, (4) constructing n-grams of term combinations (up to 3 words in one), and (5) deriving normalized word and n-gram frequencies for each document.[12]

[11] We have included one example of such a sub-query in the appendix. All other queries are available upon request.

[12] We also tried to incorporate another step (6), which added IPC dummy variables to indicate class belongings. These additional attributes where later abandoned by the following feature selection process, which suggests that these IPC class belongings are not significant for the categorization at hand.

With these normalized frequencies a matrix was constructed with columns being variables and rows being their observations. This matrix, shown in table 1, together with the binary vector indicating which observations had been identified as SR patents, served as a training input for the machine learning approach.

Table 1: *Structure of patent word and n-gram frequency matrix with binary decisions as input for machine training. The lighter gray shaded area indicates an example of a subsample, on which the machine is trained. The darker gray area is then a respective example for a subset of data which is used for testing the fitness of the classification process. The non-shaded area at the bottom refers to new data, on which the SVM is able to decide based on the previous training.*

	Attribute vectors **x**									binary
patent	$word_{w1}$	$word_{w2}$	...	$bigram_{b1}$	$bigram_{b2}$	...	$trigram_{t1}$	$trigram_{t2}$	...	decision y
1	$freq._{1\|w1}$	$freq._{1\|w2}$	...							1
2	$freq._{2\|w1}$	...								-1
...	...									...
205	$freq._{205\|w1}$	...			$freq._{205\|b2}$	...				-1
206	$freq._{206\|w1}$	...			...					1
...	...									...
228	$freq._{228\|w1}$	...						$freq._{228\|t2}$	...	-1
xxx	$freq._{xxx\|w1}$	...						...		?
xxx	$freq._{xxx\|w1}$	...								?
...	...									...

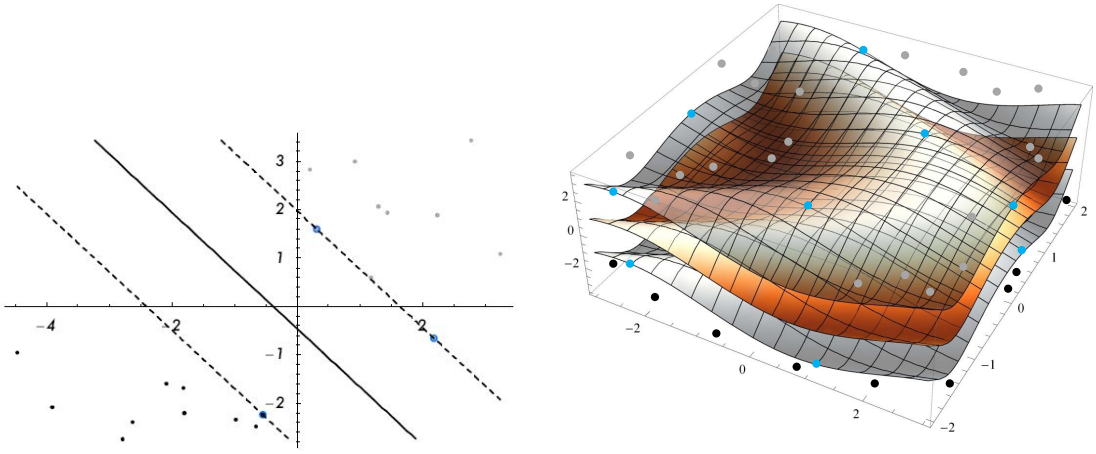
Machine Learning for Classification Analyses

Statistical classification using machine learning algorithms has long been implemented for the purpose of solving various problems and tasks such as computer vision, drug discovery or handwrite and speech recognition. Numerous different methods were developed and new ones still appear. However, there has been no one, at least to our knowledge, using statistical classifiers on the basis of a primal lexical query for the purpose of detecting an emerging technology. We considered a number of alternatives (Kotsiantis 2007) to the aforementioned SVM, such as k-Nearest Neighbor, Neural Networks, and Genetic Algorithms before starting with our particular algorithm. According to the so called *no-free-lunch* theorem (Wolpert and Macready 1997) there is no general superior machine learning method and every problem has to be tackled individually depending on its properties. We therefore assessed the aforementioned algorithms according to run-time performance, sensitivity to irrelevant or redundant features, and ability to overcome local maximums. In a nutshell, SVM proved to be the most suitable algorithm and this decision

was in line with computer science experts' opinions from robotics groups at the Karlsruhe Institute of Technology (KIT).

Support Vector Classification

The method of support vectors was introduced in the middle of the 1960s (Guyon et al. 1993, Cortes and Vapnik 1995). The original approach together with its various extensions is now one of the most acknowledged and recommended tools among modern machine learning algorithms. In the following we briefly describe its core concept and discuss some advantages that are found relevant for the problem at hand. The core idea of the method is, simply put, to create a unique discrimination profile (represented by a linear function) between samples from (usually two[13]) different classes – as depicted for a two-dimensional space in figure 1(a).



(a) linear (b) non-linear

Figure 1: *Working principle of a Support Vector Machine for linearly separable data (a) illustrated by Nilsson et al. (2006) for two dimensions, and linearly non-separable data (b) for examplarily chosen three dimensions in an own illustration. Note the optimal separating decision planes in the middle and support vectors (circled in blue). In both cases, the width of the corridor defined by the two dotted lines (a) or outer planes respectively (b) connecting the support vectors represents the margin of the optimal separating decision plane. In case of text classification axes represent normalized frequency of keyword's appearance*

The result is a line – or more generally a hyperplane – which is constructed in such a way that the distance between two parallel hyperplanes touching nearest samples becomes as large as possible. This way the method is trying to minimize false classification decisions.

[13]There exist some multiclass SVM approaches. See Duan and Keerthi (2005) for a review.

The "touching" data points are termed *support vectors*. In fact, the resulting separation plane is shaped only by these constraining (= supporting) points. Below we provide the mathematical notation of a support vector machine following Hsu et al. (2010), an article which has to be recommended as a comprehensive introduction to the method for purposes such as ours. Formally defined, we have a training set (x_i, y_i) of $i = 1, \ldots, l$ sample points (here: our patents), where every $\mathbf{x}_i \in \mathbb{R}^n$ is an attribute vector (consisting of our normalized word and n-gram frequencies) and $y_i \in \{-1, 1\}^l$ is a decision for that specific data point which thus defines its class. The SVM then yields the solution to the following optimization problem (Boser et al. 1992, Guyon et al. 1993):

$$\begin{aligned} \min_{w,b,\xi} \quad & \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{l}\xi_i \\ s.t. \quad & y_i\left(\mathbf{w}^T\Phi(\mathbf{x}_i) + b\right) \geq 1 - \xi_i \\ & \xi_i \geq 0 \end{aligned} \tag{1}$$

in which w is the normal vector between the separating hyperplane and the parallel planes spanned by the support vectors. The mapping Φ is related to so called Kernel functions, such that $K(\mathbf{x}_i, \mathbf{x}_j) \equiv \Phi(\mathbf{x}_i)^T\Phi(\mathbf{x}_j)$. For problems in which the data under consideration are not linearly separable (compare figure 1(b)), Φ maps the training attributes into a higher dimensional space where a hyperplane may be found. Table 2 summarizes common Kernel functions and their respective parameters γ, r, and d (Burges 1998, Ali and Smith-Miles 2006, Pedregosa et al. 2011, Manning et al. 2008)[14].

Table 2: *Kernel functions used for the SVM*

Kernel function	Formula
Polynomial	$(\gamma\langle x, x'\rangle + r)^d$
Radial basis function (rbf)	$\exp(-\gamma\lvert x - x'\rvert^2)$
Sigmoid	$\tanh(\langle x, x'\rangle + r)$

The above version of the classification procedure also incorporates the so called *Soft-Margin* method (Cortes and Vapnik 1995) that allows for mislabeled training sample points. The approach introduces ξ_i as non-negative slack variables which measure the extent of incorrectly classified items in the training set. $\sum_{i=1}^{l}\xi_i$ is thus a penalty term, and C a penalty parameter, on which we will comment later.

[14] Since there is no possibility to determine in advance which Kernel function should be used, the choice of the depicted functions was mostly motivated by their popularity in classifiers and availability within the software package used.

Training Algorithm, Classification, and Evaluation

Figure 2 depicts the flow chart of our algorithm. First, we preprocessed the data in order to eliminate irrelevant features and to obtain a final dataset of feature vectors. When we turn to the result section, the necessity of this preprocessing becomes clearer. In a second step we started the SVM training process. It comprised three iterative steps that are found in almost any machine learning approach: training of the model, its evaluation and optimization. We realized all these steps for our SVM using the python programming language and its tool python *scikit*-learn for machine learning (Pedregosa et al. 2011).[15] Finally, the classifier with the best model fit was applied on some test data.

The algorithm, first, randomly splits the training dataset X into training and test parts. Second, it fits the model based on the training dataset leaving out the test data. During the training process the data are again split into k parts. The algorithm then trains the model on k-1 parts and validates on the k-th part. The training is performed several times so that every part serves as a validation dataset. The number of training repetitions is reflected by a cross-validation parameter and can be specified. Thus, it is subject to variation during the overall fitting of the model itself. Figure 3 illustrates the k-fold cross-validation process.

The evaluation of our model is based on the criteria of precision and recall. The former measures the ability of a classifier not to label objects as positive that should have been labeled negative. Formally, precision is the total number of true positives (tp) divided by the sum of all positives including false positive errors (fp).

$$precision = \frac{tp}{tp + fp} \tag{2}$$

The latter, recall, measures the ability of a classifier to find all positives or, again more formally, the number of true positives divided by a sum of true positives and false negative errors (fn).

$$recall = \frac{tp}{tp + fn} \tag{3}$$

On the one hand, a model with a good recall but bad precision will find all positive samples – but will have some of them being actually negative. On the other hand, a model with bad recall but high precision will not have false positive objects, however it will miss some

[15] We do not discuss the exact implementation of the support vector machine algorithm in the python scikit-learn tool. All necessary materials can be found in open access libraries following the reference provided above. However, the appendix to this paper provides our own script written in IPython Notebook (Pérez and Granger 2007).

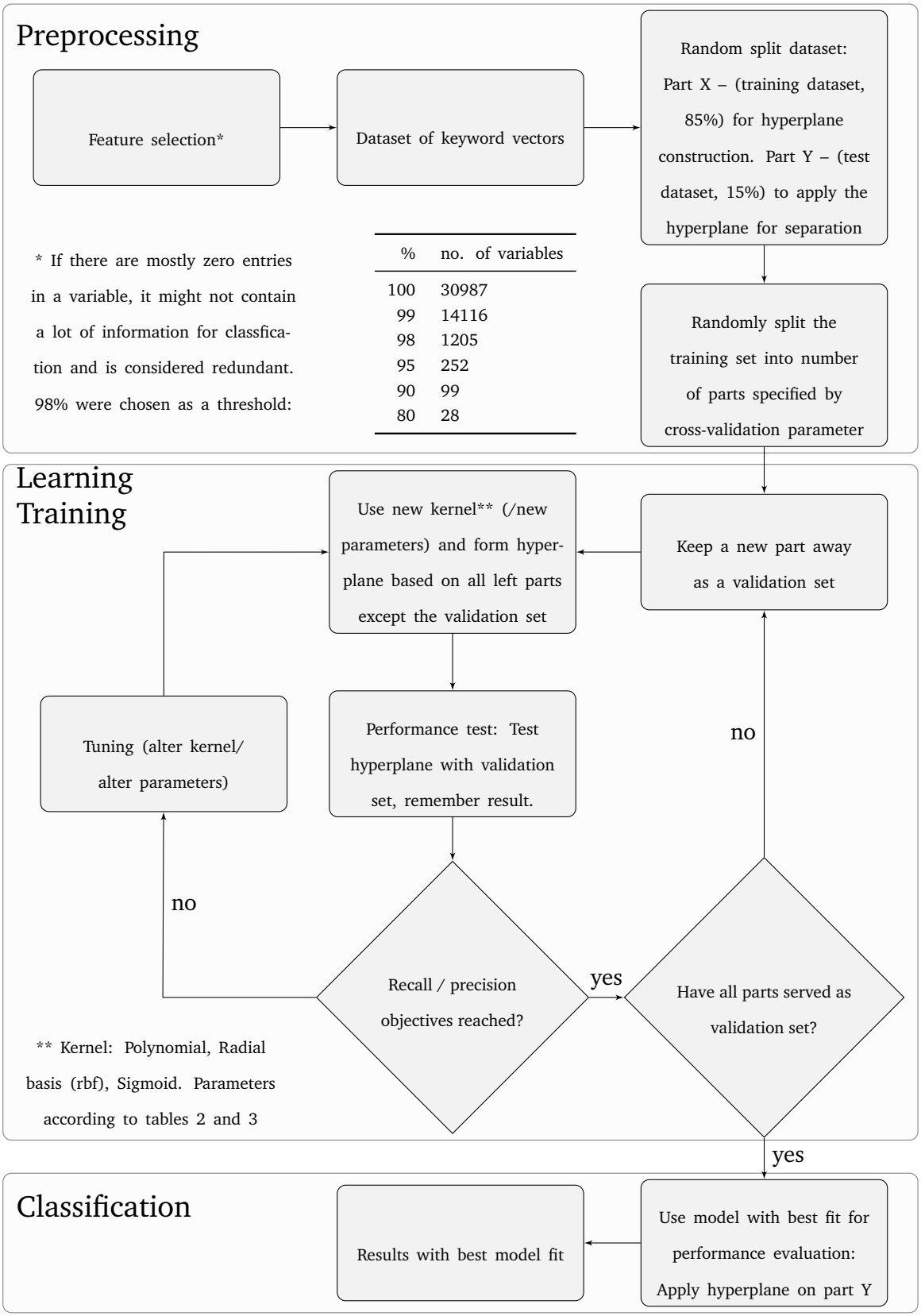


Figure 2: *Flow chart of the machine discrimination algorithm with preprocessing, support vector training, and final classification.*

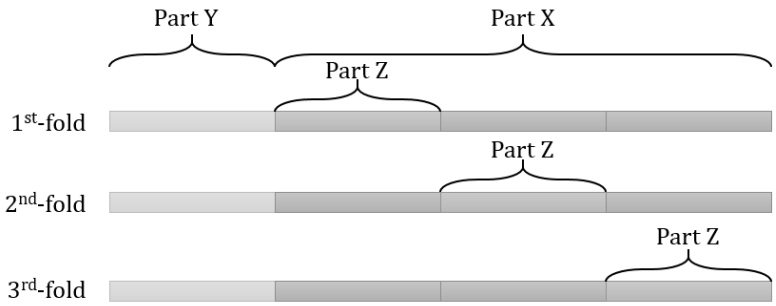


Figure 3: *k-fold cross-validation process*

of the true positives. In order to balance these two measures we used a so called f1-score that can be seen as their weighted average:

$$f1 = 2 \cdot \frac{precision * recall}{precision + recall} \tag{4}$$

To optimize our classifier we calibrated it to have the highest possible f1-score. Tuning of the model was done by varying the cross-validation parameter, the kernel functions, and their respective parameters.

4 Results

The sample used in the machine learning process consisted of 228 patents with valid expert decisions. It contained 98 SR patents and 130 IR patents according to our expert group's validation. As a result of the transformation of unstructured patent text into structured data we observed 30,987 different features (or variables) within these patents, which included key-words, bigrams, and trigrams.[16].

The resulting matrix (228 x 30,987) had to be preprocessed before serving as an input for the SVM, due to the fact that the majority of the variables contained zero entries. This means that only a small number of key-words and n-grams are shared by a majority of the patents. At first glance this information could appear confusing. The explanation lies in the variety of SR applications: Descriptions of significantly different service robots with very unlike applications contain a huge number of dissimilar key-words and key-word combinations. Most of these are uniquely used in their specific contexts and thus appear with a very low frequency. Figure 4 illustrates this fact by showing typical relative appearances of normalized frequencies of four randomly chosen variables.

[16]We even included IPC classes in an early stage of development, but did not find any of these classifications to become part of the support vectors. They turned out to irrelevant for the discrimination procedure and were thus removed during the feature selection process

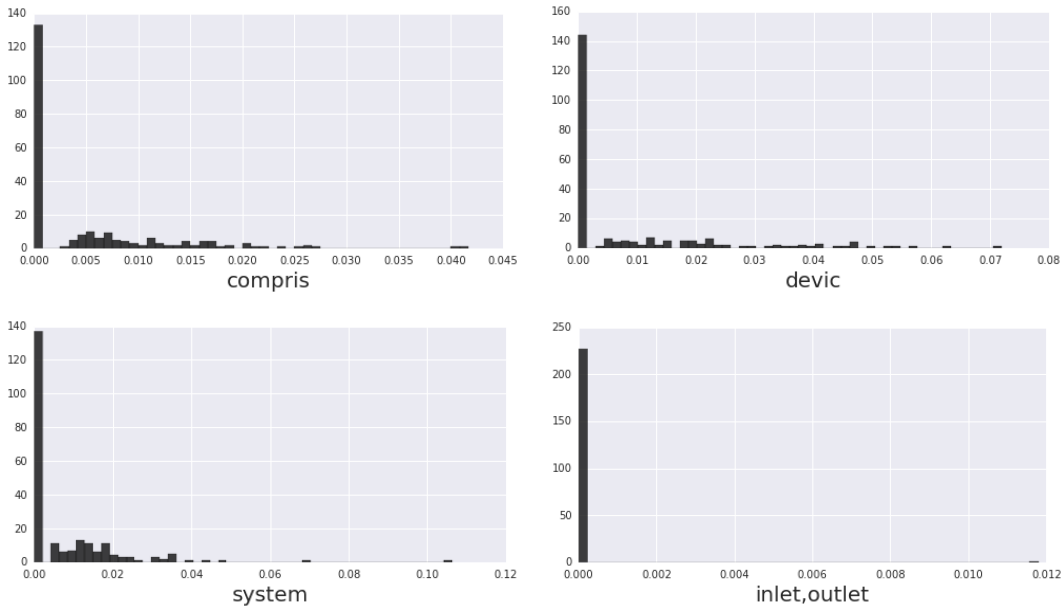


Figure 4: *Four histograms of random key-words and one bigram.*

Thus some variables contained too little information and introduced noise instead. Consequently, these insignificant features had to be excluded from the data – for the purpose of improving the SVM performance. For example, if a key-word (or n-gram) appeared in only one patent, this variable would not have helped in solving the problem of classification. Our feature selection process served to exclude such a redundant feature. We implemented a threshold that at least 2% of the entries of a variable in each class (SR vs. IR) should have non-zero entries. The table in the flow chart (figure 2) shows the dependency between the number of variables and different thresholds. With this selection process the resulting matrix was reduced to 1,206 variables for our 228 observations/patents. We provide these variables/terms in the tables 10 to 16 in the appendix. Finally all variable frequencies were scaled to the interval $[0, 1]$ such that a second normalization process set the maximum frequency in the sample to 1.

Figures 5 and 6 show normalized frequencies of attribute pairs and groups of three respectively. Coloured dots indicate the expert classification as SR (red) and IR (blue).

SVM specific outcomes

In order to eliminate negative influence of the unbalanced dataset we introduced weights in our SVM proportionate to SR and IR classes. Following the cross-validation procedure the support vector machine was fit on to a 85% of the original dataset. The remaining 15% were kept for testing purposes. The split was random and its ratio is an arbitrary choice of authors.

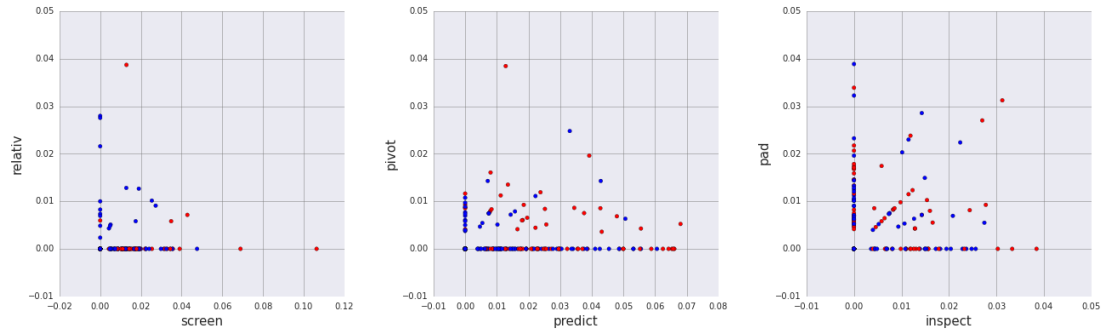


Figure 5: *Normalized frequencies of randomly chosen attribute pairs – here key-words. Coloured dots indicate the expert classification as SR (red) and IR (blue).*

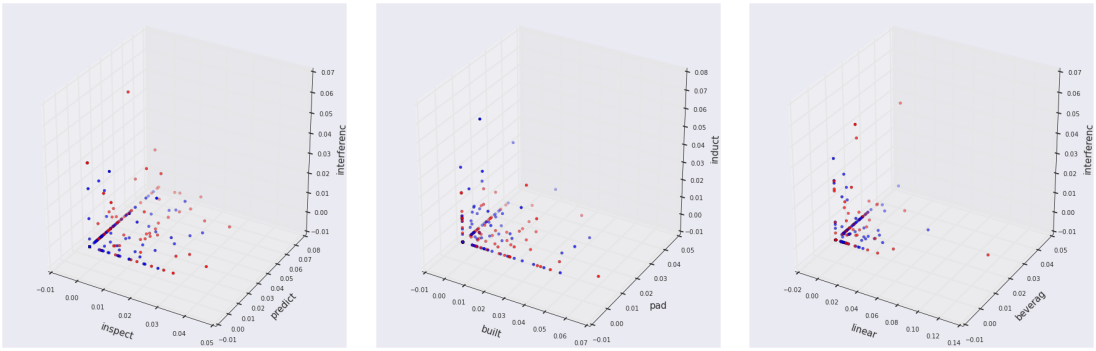


Figure 6: *Normalized frequencies of randomly chosen attribute groups of three – here key-words. Coloured dots indicate the expert classification as SR (red) and IR (blue).*

The cross-validation parameter was set to 3 and 4 determining the amount of random splits of a training dataset into a training and evaluation set. Another parameter that was varied while searching for a better model is so called C parameter. The following citation nicely explains the main properties of this penalty parameter: "*In the support-vector network algorithm one can control the trade-off between complexity of decision rule and frequency of error by changing the parameter C*" (Cortes and Vapnik 1995, p. 286).

Finally, the three different kernel functions from table 2 were considered. In particular, the first was a polynomial function and its γ, degree, and r coefficient. The second was a radial basis function (rbf) and its γ constant. The third was a sigmoid function and its γ and r constant. Table 3 presents all kernel parameters and their values that were considered to find the best performing classifier – as well as all eventually chosen values.

Exhaustive simulations with all possible combinations of the above mentioned parameters yielded the best f1-score of the model. Our final model showed an 85% precision and 83% recall. It contained a radial basis function kernel with γ equal to 0.005 and C equal to 10. The training set was randomly split into 3 equal parts for cross validation. The

Table 3: *Model tuning parameters and respective values*

Parameter	Varied values	Chosen values
cross-validation (cv)	$3,4$	3
complexity (C)	$10,\ldots,1000$	10
γ of rbf kernel	$10^{-6},\ldots,10^{-2}$	0.005
γ of polynomial kernel	$10^{-6},\ldots,10^{-2}$	not chosen
d of polynomial kernel	$1,2,3$	not chosen
r of polynomial kernel	$1,2,3$	not chosen
γ of sigmoid kernel	$10^{-6},\ldots,10^{-2}$	not chosen
r of sigmoid kernel	$1,2,3$	not chosen

resulting discrimination plane between the two classes of patents was constructed using 192 support vectors, meaning that only these sample observations were significant for classification. Table 4 presents a classification report after classifying the test set of our sample.

Table 4: Classification report

	precision	*recall*	*f1-score*	*No. of patents in test set*
SR	75%	94%	83%	16
IR	93%	74%	82%	19
Avg. / total	85%	83%	83%	35

Scope for Improvement

There is some scope for an even more precise technology identification. First, there is still room to increase the performance of the SVM method, namely regarding the kernel functions. Although there have not been any successful attempts to introduce automatic kernel selection algorithms yet (Ali and Smith-Miles 2006), it is probably possible to find a better function for our problem at hand. Second, the support vector machine can be seen as a first-tier machine classifier that we just started with. Other methods like genetic algorithms, neural networks or boosting as well as their combinations could be applied in additional steps. Finally, applying principal component analyses (PCA) to our matrix of variables could provide some insights about a similar behavior of different key words in patents, which means that they could be grouped and analyzed together. Moreover, applying PCA in SVM we could say whether these groups of variables are significant in identifying an emerging technology – like service robotics in our show case.

5 Conclusion

In this paper we proposed a novel methodology for detecting early developments of an emerging technology in patent data. Our method uses a support vector machine algorithm on the example of robotics patents. The resulting model is able to find 83% of service robotics patents and classify them correctly with a probability of 85%.

There are several advantages of our method regarding technology classification tasks, which we will discuss along the criteria of Mogoutov and Kahane (2007) that we mentioned above: First, experts do not choose which terms should be added to or excluded from the primal search, hence the typical lexical bias towards preferred subfields is limited. Speaking of lexical versus citationist apporaches, our method also avoids a major drawback of citational methods which circle around a core dataset and rely on future works explicitly referring to this prior art: Since citations in patents are generally rare[17], for young emerging technologies in particular the citation lag decreases the expected number of citations for any given document to a negligible amount. Second, the procedure offers strong portability, such that it can easily be applied to scientific publications - taken for instance from Web of Science. Moreover, our step-by-step classification method can basically be applied to any emerging technology - not only those that arise as an initially small subset consisting of niche applications like SR emerging out of Robotics. Nanotechnology, which in this respect is again a meaningful instance, would have been hard to detach from some well-defined mother technology. In fact, it became an umbrella term for technological developments from various directions that had solely in common to work on a sufficiently small scale and to make intentionally use of the phenomena that arise on this scale. Nanotechnology thus consolidated endeavours from physics, chemistry, material technologies and biology and had a converging character. The same is true for Industry 4.0, which is a superordinate concept describing digitally cross-linked production systems and thus enveloping various heterogenous sub-technologies that are hardly classifiable. One of our future tasks will thus comprise the application of our method on historical nanotechnological patent sets as well as on Industry 4.0 technologies in order to demonstrate the general applicability and robustness of our method. Third, our algorithm approach shows high adaptability. Due to its learning nature it is able to produce valid outcomes although the technology under consideration is constantly evolving. Fourth and of capital importance, the proposed method performs well in terms of recall and precision scores, proving sufficient extent and relevance of the obtained data.

[17] Within PATSTAT, for instance, more than 90% of the listed patent applications are followed by less than three forward citations, 74% do not show any at all.

Appendix

Table 5: *Important robot definitions according to ISO 8373:2012*

	Definition
Robot:	Actuated mechanism programmable in two or more axes with a degree of autonomy, moving within its environment, to perform intended tasks. Note 1 to entry: A robot includes the control system and interface of the control system. Note 2 to entry: The classification of robot into industrial robot or service robot is done according to its intended application.
Autonomy:	Ability to perform intended tasks based on current state and sensing, without human intervention.
Control System:	Set of logic control and power functions which allows monitoring and control of the mechanical structure of the robot and communication with the environment (equipment and users).
Robotic Device:	Actuated mechanism fulfilling the characteristics of an industrial robot or a service robot, but lacking either the number of programmable axes or the degree of autonomy.

Table 6: *SR application examples for personal / domestic use according to the IFR*

	Applications
Robots for domestic tasks	Robot butler, companion, assistants, humanoids Vacuuming, floor cleaning Lawn mowing Pool cleaning Window cleaning
Entertainment robots and Toy robots	Robot rides Pool cleaning Education and training
Handicap assistance and Robotized wheelchairs	Personal rehabilitation Other assistance functions
Personal transportation	
Home security and surveillance	

Table 7: *SR application examples for professional / commercial use according to* **?**

	Applications
Field robotics	Agriculture
	Milking robots
	Forestry
	Mining systems
	Space robots
Professional cleaning	Floor cleaning
	Window and wall cleaning
	Tank, tube and pipe cleaning
	Hull cleaning
Inspection and maintenance systems	Facilities, Plants
	Tank, tubes and pipes and sewer
	Other inspection and maintenance systems
Construction and demolition	Nuclear demolition and dismantling
	Other demolition systems
	Construction support and maintenance
	Construction
Logistic systems	Courier/Mail systems
	Factory logistics
	Cargo handling, outdoor logistics
	Other logistics
Medical robotics	Diagnostic systems
	Robot assisted surgery or therapy
	Rehabilitation systems
	Other medical robots
Defense, rescue and security applications	Demining robots
	Fire and bomb fighting robots
	Surveillance/security robots
	Unmanned aerial and ground based vehicles
Underwater systems	Search and Rescue Applications
	Other
Mobile Platforms in general use	Wide variety of applications
Robot arms in general use	Wide variety of applications
Public relation robots	Hotel and restaurant robots
	Mobile guidance, information robots
	Robots in marketing
Special Purpose	Refueling robots
Customized robots	Customized applications for consumers
Humanoids	Variety of applications

Table 8: *Exemplary extract of robot patents under consideration with respective titles, publication numbers (given by the patent authority issuing the patent), filing dates (on which the application was received), and expert classifcation decisions*

Title	Publication no.	Filing date	SR y/n?
Remote control manipulator	968525	1962-06-25	n (-1)
Folded robot	2061119	1979-10-24	n (-1)
In vivo accessories for minimally invasive robotic surgery	2002042620	2001-11-06	y (1)
Apparatus and method for non-destructive inspection of large structures	6907799	2001-11-13	y (1)
Surgical instrument	2002128661	2001-11-16	y (1)
Robotic vacuum cleaner	2003060928	2001-12-04	y (1)
A cleaning device	1230844	2002-01-21	n (-1)
Climbing robot for movement on smooth surfaces e.g. automatic cleaning of horizontal / vertical surfaces has chassis with crawler drive suspended and mounted turnable about vertical axis, to detect obstacles and prevent lifting-off	10212964	2002-03-22	y (1)
Single Cell Operation Supporting Robot	2004015055	2002-08-08	y (1)
Underwater Cleaning Robot	2007105303	2006-03-14	y (1)
Position determination for medical devices with redundant position measurement and weighting to prioritise measurements	1854425	2006-05-11	y (1)
Mobile Robot and Method of controlling the same	2007135736	2006-05-24	y (1)
Customizable Robotic System	2012061932	2011-11-14	y (1)
Positioning Apparatus for Biomedical Use	2012075571	2011-12-06	n (-1)
Apparatus and Method of Controlling Operation of Cleaner	2012086983	2011-12-19	n (-1)

Table 9: *Modular SQL Boolean term search approach for PATSTAT, defined through specific word construction for IFR application field* CLEANING SR*, augmented by IPC class codes. AST refers to table containing abstracts, TTL refers to table containing titles, IPC refers to table containing IPC classes.*

```
(
        (
        SUBSTRING(IPC.ipc_class_symbol,1,5)='"B08B'     OR
        SUBSTRING(IPC.ipc_class_symbol,1,5)='"E01H'     OR
        IPC.ipc_class_symbol LIKE '%B60S 1%'            OR
        IPC.ipc_class_symbol LIKE '%B60S 3%'
        )
OR (
        TTL.appln_title LIKE '%robot%'                  AND (
                                                                TTL.appln_title LIKE '%suction cup%'        OR
                                                                TTL.appln_title LIKE '%safety analy%'       OR
                                                                TTL.appln_title LIKE '%vertical wall%'      OR
                                                                TTL.appln_title LIKE '%dry adhesive%'       OR
                                                                TTL.appln_title LIKE '%gecko%'              OR
                                                                TTL.appln_title LIKE '%wheel-based%'        OR (
                                                                TTL.appln_title LIKE '%clean%'              AND NOT (   TTL.appln_title LIKE '%house%'      OR
                                                                                                                        TTL.appln_title LIKE '%domestic%'   OR
                                                                                                                        TTL.appln_title LIKE '%pool%'
                                                                                                            ))
                                                                                                            OR (
                                                                TTL.appln_title LIKE '%climb%'              AND NOT     TTL.appln_title LIKE '%wheelchair%'
                                                                                                            )
                                                )
                                                AND NOT (       TTL.appln_title LIKE '%vacuum%'             OR
                                                                AST.appln_abstract LIKE '%vacuum%'          OR
                                                                TTL.appln_title LIKE '%wafer%'              OR
                                                                AST.appln_abstract LIKE '%wafer%'           OR
                                                                TTL.appln_title LIKE '%semiconductor%'      OR
                                                                AST.appln_abstract LIKE '%semiconductor%'   OR
                                                                AST.appln_title LIKE '%industr%'            OR
                                                                AST.appln_abstract LIKE '%industr%'         OR
                                                                TTL.appln_title LIKE '%milk%'               OR
                                                                AST.appln_abstract LIKE '%milk%'            OR
                                                                TTL.appln_title LIKE '%paint%'              OR
                                                                AST.appln_abstract LIKE '%paint%'           OR
                                                                TTL.appln_title LIKE '%weld%'               OR
                                                                AST.appln_abstract LIKE '%weld%'            OR
                                                                TTL.appln_title LIKE '%manufact%'           OR
                                                                AST.appln_abstract LIKE '%manufact%'
                                                )
)
OR (
        AST.appln_abstract LIKE '%robot%'               AND (   AST.appln_abstract LIKE '%suction cup%'     OR
                                                                AST.appln_abstract LIKE '%safety analy%'    OR
                                                                AST.appln_abstract LIKE '%vertical wall%'   OR
                                                                AST.appln_abstract LIKE '%dry adhesive%'    OR
                                                                AST.appln_abstract LIKE '%gecko%'           OR
                                                                AST.appln_abstract LIKE '%wheel-based%'     OR (
                                                                AST.appln_abstract LIKE '%clean%'           AND NOT (   AST.appln_abstract LIKE '%house%'   OR
                                                                                                                        AST.appln_abstract LIKE '%domestic%'OR
                                                                                                                        AST.appln_abstract LIKE '%pool%'
                                                                                                            ))
                                                                                                            OR (
                                                                AST.appln_abstract LIKE '%climb%'           AND NOT     AST.appln_abstract LIKE '%wheelchair%'
                                                                                                            )
                                                )
                                                AND NOT (       TTL.appln_title LIKE '%vacuum%'             OR
                                                                AST.appln_abstract LIKE '%vacuum%'          OR
                                                                TTL.appln_title LIKE '%wafer%'              OR
                                                                AST.appln_abstract LIKE '%wafer%'           OR
                                                                TTL.appln_title LIKE '%semiconductor%'      OR
                                                                AST.appln_abstract LIKE '%semiconductor%'   OR
                                                                AST.appln_title LIKE '%industr%'            OR
                                                                AST.appln_abstract LIKE '%industr%'         OR
                                                                TTL.appln_title LIKE '%milk%'               OR
                                                                AST.appln_abstract LIKE '%milk%'            OR
                                                                TTL.appln_title LIKE '%paint%'              OR
                                                                AST.appln_abstract LIKE '%paint%'           OR
                                                                TTL.appln_title LIKE '%weld%'               OR
                                                                AST.appln_abstract LIKE '%weld%'            OR
                                                                TTL.appln_title LIKE '%manufact%'           OR
                                                                AST.appln_abstract LIKE '%manufact%'
                                                )
)
)
AND NOT     SUBSTRING (IPC.ipc_class_symbol,1,4)='"A47'
```

Table 10: List of the 1,206 variables used in the SVM for classification: Part 1/4 of the 726 unigrams.

1a	arrang	cardiac	confirm
abl	arrangement	carri	connect
abnormal	arriv	carriag	connection
accelerat	articulat	carrier	consequent
access	assembl	caus	consist
accommodat	assist	cell	constitut
accord	associat	center	construct
accordanc	attach	centr	construction
accurat	attachabl	central	contact
achiev	attachment	chang	contain
acquir	auto	characteris	container
act	automat	characteristic	continuous
action	automatic	characteriz	control
activ	autonomous	charg	controller
actual	auxiliari	chassi	convention
actuat	avoid	check	convert
adapt	axe	circuit	conveyor
adapter	axi	claim	coordinat
addition	axial	clamp	correspond
adhesiv	backlash	clean	cost
adjacent	balanc	cleaner	coupl
adjust	barrier	climb	cover
adjustabl	base	clip	creat
adjustment	basi	close	crop
advanc	beam	coat	current
advantag	bear	code	customizabl
agricultural	behavior	collect	cut
aid	bend	collision	damag
aim	bicycl	column	data
air	bipedal	combin	decision
algorithm	blade	combinat	defin
allow	block	comfortabl	degre
amount	board	command	deliver
analysi	bodi	common	deliveri
analyz	bore	communic	deploy
angl	bottom	compact	depress
angular	box	compar	describ
animal	brush	compartment	design
annular	build	complementari	desir
apertur	built	complet	detachabl
apparatus	button	component	detect
appearanc	cabl	compos	detection
appli	calculat	compris	detector
applianc	camera	computer	determin
applic	capabl	condition	determinat
appropriat	capillari	configur	deviat
architectur	captur	configurat	devic
arm	car	confin	diagnosi

Table 11: List of the 1,206 variables used in the SVM for classification: Part 2/4 of the 726 unigrams.

differenc	endoscopic	form	inspection
difficult	energi	frame	instal
digital	engag	free	installat
dimension	enhanc	freedom	instruction
dimensional	ensur	frequenc	instrument
dip	enter	front	integrat
direct	entir	function	interaction
direction	environment	gear	interconnect
discharg	environmental	generat	interfac
disclos	equip	glove	interior
disconnect	equipment	grasp	internal
dispens	error	grip	invasiv
displac	especial	gripper	invention
displaceabl	essential	groov	involv
displacement	etc	ground	item
display	exampl	guid	jet
dispos	exchang	guidanc	join
distal	exhaust	hand	joint
distanc	exist	handl	knee
dock	expensiv	har	laser
door	extend	head	latter
doubl	extension	heat	lawn
draw	external	held	layer
drill	extract	help	leg
drive	extraction	hip	length
driven	extrem	hold	lever
dust	facilitat	holder	lift
dynamic	faciliti	horizontal	light
earth	factor	hose	limb
easili	fasten	hous	limit
edg	featur	human	line
effect	feedback	hydraulic	linear
effectiv	field	identifi	link
effector	fig	imag	liquid
efficienc	figur	implement	load
elastic	fill	improv	local
electric	filter	improvement	locat
electronic	finger	includ	lock
element	fit	incorporat	locomotion
elongat	fix	increas	log
embodiment	flang	independent	longitudinal
emit	flat	individual	loop
emitter	flexibl	industrial	low
employ	floor	informat	lower
employment	flow	inner	machin
enabl	fluid	input	magnetic
enclos	forc	insert	main
endoscop	foreign	insertion	maintain

Table 12: List of the 1,206 variables used in the SVM for classification: Part 3/4 of the 726 unigrams.

make	obtain	portion	referenc
manipulat	oper	position	region
manner	operabl	possibl	register
manoeuvr	operat	power	relat
manual	oppos	pre	relationship
manufactur	optic	precis	relativ
map	option	predefin	releas
marker	orient	predetermin	reliabl
master	orientat	preferabl	remot
material	orthogonal	preparat	remov
mean	outer	press	removal
measur	output	pressur	replac
measurement	overal	prevent	requir
mechanic	pair	procedur	resolution
mechanism	pallet	process	respect
medic	panel	processor	respectiv
medicin	parallel	produc	result
medium	part	product	retain
memori	partial	production	return
method	particular	program	rigid
micro	pass	project	ring
militari	path	propos	risk
milk	patient	propulsion	robot
mine	pattern	protectiv	robotic
minimal	payload	provid	rock
mobil	perform	proximal	rod
modal	performanc	purpos	roll
mode	period	quantiti	roller
model	peripheral	rack	rotari
modul	permit	radar	rotat
monitor	perpendicular	radial	rotatabl
motion	photograph	radio	rough
motor	pick	rail	run
mount	piec	rais	safeti
movabl	pipe	rang	sampl
move	pivot	rapid	save
movement	pivotabl	reach	scale
mow	place	reaction	screen
mower	plan	real	seal
mri	plane	realiti	section
multi	plant	realiz	sector
multipl	plastic	rear	secur
navigat	plate	receiv	select
network	platform	receiver	send
normal	play	reciprocat	sens
nozzl	plural	recognition	sensor
object	pneumatic	record	sent
obstacl	port	reduc	separat

Table 13: List of the 1,206 variables used in the SVM for classification: Part 4/4 of the 726 unigrams.

sequenc	substantial	transmission	wire
seri	substrat	transmit	wireless
serv	subsystem	transmitter	workpiec
servo	suction	transport	worn
set	suitabl	transportat	wrist
shaft	suppli	transvers	zone
shape	support	travel	
shield	surfac	treat	
ship	surgeon	treatment	
short	surgeri	tube	
signal	surgic	type	
significant	surround	typic	
simpl	sutur	ultrasonic	
simulat	switch	underwater	
simultaneous	system	uneven	
singl	take	unit	
site	tank	universal	
situat	target	unload	
size	task	upper	
skin	techniqu	use	
slave	telepresenc	user	
sleev	telescopic	utiliz	
smooth	terminal	vacuum	
sourc	terrain	valu	
sow	test	variabl	
space	therebi	varieti	
spatial	therefrom	vehicl	
special	thereof	velociti	
specifi	thereon	vertic	
specific	thereto	vessel	
speed	third	video	
spiral	tight	view	
spray	tilt	virtual	
spring	time	visual	
stabiliti	tip	volum	
stabiliz	tissu	walk	
stabl	tool	wall	
stage	tooth	wast	
station	top	water	
stationari	torqu	weed	
steer	torso	weight	
step	touch	weld	
stop	toy	wheel	
storag	track	wherebi	
store	train	wherein	
structur	trajectori	wide	
subject	transfer	winch	
subsequent	translat	window	

Table 14: List of the 1,206 variables used in the SVM for classification: Part 1/2 of the 370 bigrams.

1,2	button,effector	deviat,actual	imag,process
1,compris	capabl,control	devic,17	implement,method
1,computer	cardiac,procedur	devic,compris	includ,base
1,connect	chassi,frame	devic,control	includ,main
1,disclos	claim,includ	devic,determin	includ,pair
12,includ	clean,horizontal	devic,direct	includ,step
12,provid	clean,method	devic,includ	independent,claim
13,14	clean,operat	devic,main	industrial,robot
2,3	clean,robot	devic,position	informat,relat
2,compris	cleaner,compris	devic,provid	informat,sensor
2,move	cleaner,invention	devic,robot	informat,set
3,4	comfortabl,position	devic,system	inner,surfac
3,compris	component,provid	direction,drive	input,button
3,connect	compris,base	displacement,sensor	input,data
4,5	compris,bodi	distanc,measur	instrument,coupl
43,connect	compris,main	door,10	instrument,effector
5,arrang	compris,plural	drive,actuat	instrument,mount
5,provid	compris,robot	drive,devic	invasiv,cardiac
accord,invention	compris,robotic	drive,forc	invention,compris
actual,position	computer,program	drive,ground	invention,disclos
actuat,control	connect,clamp	drive,mechanism	invention,propos
addition,equipment	control,box	drive,system	invention,provid
adjust,position	control,cabl	drive,unit	invention,relat
adjustabl,surgeon	control,devic	drive,wheel	joint,provid
allow,surgeon	control,input	e,g	laser,emitter
angl,adjust	control,joint	effector,control	leg,joint
apparatus,compris	control,manipulat	effector,correspond	longitudinal,direction
apparatus,method	control,method	effector,handl	machin,tool
apparatus,perform	control,movement	effector,manipulat	main,bodi
arm,coupl	control,operat	effector,move	main,controller
arm,includ	control,panel	effector,movement	manipulat,arm
arm,instrument	control,provid	effector,perform	manipulat,hold
arm,join	control,resolution	element,5	master,handl
assembl,method	control,robot	endoscopic,imag	mean,14
automatic,clean	control,robotic	error,signal	mean,2
automatic,control	control,system	factor,adjustabl	mean,detect
automatic,robot	control,unit	front,bodi	mean,receiv
autonomous,move	controller,handl	front,rear	measur,devic
autonomous,robot	correspond,movement	front,robot	mechanism,rotat
axe,rotat	coupl,pair	guid,mean	method,apparatus
balanc,control	degre,freedom	hand,surgeon	method,autonomous
base,informat	deliveri,system	handl,controller	method,clean
base,station	depress,surgeon	handl,move	method,control
bodi,2	detect,obstacl	handl,scale	method,invention
bodi,robot	detect,position	har,1	method,provid
bodi,surgic	detection,mean	hold,sutur	method,system
button,allow	determin,position	horizontal,vertic	method,thereof
button,depress	determin,spatial	imag,data	method,use

Table 15: List of the 1,206 variables used in the SVM for classification: Part 2/2 of the 370 bigrams.

minimal,invasiv	position,coordinat	robot,pick	system,includ
mobil,robot	position,determinat	robot,position	system,method
mobil,robotic	position,devic	robot,realiz	system,mobil
motion,control	position,handl	robot,robot	system,perform
motion,controller	position,informat	robot,s	system,robot
motor,drive	position,robot	robot,system	system,use
motor,vehicl	position,robotic	robotic,arm	thereof,invention
mount,chassi	position,system	robotic,control	time,period
mount,robot	power,sourc	robotic,devic	tissu,robotic
move,button	predetermin,position	robotic,surgeri	travel,perform
move,comfortabl	predetermin,time	robotic,system	tube,apparatus
move,devic	procedur,system	rotari,brush	typic,movement
move,effector	produc,correspond	rotat,axe	uneven,terrain
move,floor	provid,mean	rotat,head	unit,arrang
move,robot	provid,platform	rotat,motor	unit,compris
move,surgeon	provid,robot	rotat,movement	unit,control
movement,effector	provid,surgic	rotat,shaft	unit,drive
movement,handl	purpos,robot	scale,effector	unit,generat
movement,movement	real,time	scale,factor	unit,provid
movement,perform	relat,automatic	seal,access	upper,lower
movement,robotic	relat,method	send,imag	use,robotic
movement,typic	relat,mobil	sensor,mount	use,surgic
navigat,system	relat,robot	servo,motor	user,operat
object,provid	remot,control	signal,receiv	vacuum,clean
operat,accord	remot,view	signal,robot	vacuum,cleaner
operat,clamp	resolution,effector	signal,transmitter	vehicl,bodi
operat,devic	robot,1	slave,robot	vertic,axi
operat,operat	robot,10	smooth,surfac	video,signal
operat,perform	robot,arm	sow,weed	walk,robot
operat,power	robot,arrang	surfac,clean	water,discharg
operat,rang	robot,automatic	surgeon,adjust	wheel,instal
operat,remot	robot,bodi	surgeon,control	wire,wireless
operat,robot	robot,capabl	surgeon,input	x,y
operat,unit	robot,clean	surgeon,produc	y,z
output,signal	robot,cleaner	surgeon,scale	
overal,structur	robot,communic	surgeri,surgic	
pair,master	robot,compris	surgic,instrument	
pair,robotic	robot,control	surgic,operat	
pair,surgic	robot,includ	surgic,procedur	
path,robot	robot,invention	surgic,robot	
patient,s	robot,main	surgic,site	
patient,treat	robot,method	surgic,system	
perform,clean	robot,mobil	surgic,tool	
perform,hand	robot,motion	sutur,tissu	
perform,minimal	robot,move	system,autonomous	
perform,surgic	robot,movement	system,compris	
position,base	robot,mower	system,control	
position,compris	robot,operat	system,devic	

Table 16: List of the 1,206 variables used in the SVM for classification: All 110 trigrams.

adjust,position,handl	invention,relat,automatic	surgeon,produc,correspond
adjustabl,surgeon,control	invention,relat,method	surgeon,scale,factor
allow,surgeon,adjust	invention,relat,mobil	surgic,instrument,coupl
apparatus,perform,minimal	manipulat,hold,sutur	surgic,instrument,mount
arm,coupl,pair	master,handl,controller	surgic,robot,compris
arm,instrument,effector	method,invention,relat	surgic,robot,system
button,allow,surgeon	method,thereof,invention	sutur,tissu,robotic
button,depress,surgeon	minimal,invasiv,cardiac	system,control,movement
button,effector,move	mobil,robot,invention	system,includ,pair
cardiac,procedur,system	mobil,robotic,devic	system,perform,minimal
clean,horizontal,vertic	mount,robot,arm	thereof,invention,disclos
clean,robot,1	move,button,depress	tissu,robotic,arm
cleaner,invention,relat	move,comfortabl,position	typic,movement,perform
compris,main,bodi	move,effector,handl	x,y,z
control,input,button	move,surgeon,produc	
control,method,thereof	movement,effector,control	
control,resolution,effector	movement,effector,movement	
controller,handl,move	movement,handl,scale	
correspond,movement,effector	movement,movement,effector	
correspond,movement,typic	movement,perform,hand	
coupl,pair,master	movement,typic,movement	
coupl,pair,robotic	pair,master,handl	
depress,surgeon,input	pair,robotic,arm	
devic,main,controller	pair,surgic,instrument	
devic,robot,arm	perform,clean,operat	
effector,control,input	perform,hand,surgeon	
effector,correspond,movement	perform,minimal,invasiv	
effector,handl,move	position,handl,move	
effector,manipulat,hold	position,robot,arm	
effector,move,button	procedur,system,includ	
effector,movement,handl	produc,correspond,movement	
effector,movement,movement	relat,automatic,robot	
factor,adjustabl,surgeon	resolution,effector,movement	
front,robot,arm	robot,arm,includ	
hand,surgeon,scale	robot,cleaner,compris	
handl,controller,handl	robot,cleaner,invention	
handl,move,comfortabl	robot,control,method	
handl,move,effector	robot,control,system	
handl,move,surgeon	robot,invention,relat	
handl,scale,effector	robot,system,method	
hold,sutur,tissu	robotic,arm,coupl	
includ,pair,surgic	robotic,arm,instrument	
independent,claim,includ	robotic,devic,compris	
input,button,allow	scale,effector,correspond	
input,button,effector	scale,factor,adjustabl	
instrument,coupl,pair	surgeon,adjust,position	
instrument,effector,manipulat	surgeon,control,resolution	
invasiv,cardiac,procedur	surgeon,input,button	

Figure 7: *The following python scripts were used to implement the support vector machine, load the preprocessed text elements, and select relevant attributes for classification.*

In [ ]:
```
from collections import defaultdict
import itertools
from sklearn import svm
from scipy import stats
import numpy as np
import seaborn as sns
import scipy as sp
from random import randint
from sklearn.cross_validation import train_test_split
from sklearn.grid_search import GridSearchCV
from sklearn.metrics import classification_report
from sklearn.metrics import precision_score
from sklearn.metrics import f1_score
from sklearn.metrics import recall_score
from sklearn.svm import SVC
```

Load information

In [ ]:
```
def transform_str(string):
    if string <> '0' and string <> '0\r\n':
        number=float(string.split('/')[0])/float(string.split('/')[1])
    else:
        number=0
    return number

Pat_vectors=list()
#Loading Key_word vectors
with open(r'6_Sample\141119_fVectorAllNormalized_NoGerman.csv', 'rb') as f1:
    for line in f1:
        Patent_vector=list()
        for element in line.split(','):
            Patent_vector.append(transform_str(element))
        Pat_vectors.append(Patent_vector)
X=np.array(Pat_vectors)
y=list()
#Loading decisions
with open(r'6_Sample\141119_decisionVector_NoGerman.csv', 'rb') as f2:
    for line in f2:
        y.append(float(line.strip()))
print 'Number of patents/ observations ', len(X)# X.size
print 'Number of key-words/ variables = ', len(X[0])
```

Feature Selection (Dimension reduction)

In [ ]:
```
def feature_selection(X,y,Features_names,Threshold):
    '''
    A function selects features based on the number of zeros in the variable relative to different classes.
    Or how chaotic zeros are distributed among classes in the variable.
    X numpy array like (n_samples,n_features)
    y numpy array like (n_samples) decision vector

    returns X_modified numpy array like (n_samples,n_features_selected) with only selected features
    - N1>>N2 and N3>>N4 Mostly 0 as entries of the variable  (Lack of information in the variable)
    + N1>>N2 and N3<<N4 Mostly 0 in 1st class and >0 in 2nd  (Determinant of 2nd Class)
    + N1<<N2 and N3>>N4 Mostly 0 in 2nd class and >0 in 1st  (Determinant of 1st Class)
    + N1<<N2 and N3<<N4 Mostly >0 as entries of the variable (Can't conclude anything)

    >> or << is determined by a local 'Threshold' variable
    The probability that a given word (n-gramm,...) appears in any class less then that a threshold
    '''
    List=list()
    kw_list=list() # list of the key words that are selected for classification
    col_num=0
    for col in X.T:
        N1,N2,N3,N4 =0,0,0,0
        for v in range(len(col)):
            if col[v]>0:
                if y[v]==1: N2+=1
                else: N4+=1
            else:
                if y[v]==1: N1+=1
                else: N3+=1
        try:
            if N2/float(N1)<Threshold and N4/float(N3)<Threshold:
                pass
            else:
                kw_list.append(Features_names[col_num])
                List.append(col)
        except ZeroDivisionError:
            kw_list.append(Features_names[col_num])
            List.append(col)
        col_num+=1
    X_mod=np.array(List)
    X_mod=X_mod.T
    return (X_mod,kw_list)

Threshold=0.02
X_modified=feature_selection(X_,y,New_features,Threshold)[0]
list_of_kw=feature_selection(X_,y,New_features,Threshold)[1]
print 'Number of variables left',len(list_of_kw)
print 'Number of patents/ observations ', len(X_modified)# X_modified.size
print 'Number of key-words/ variables = ', len(X_modified[0])
```

Figure 8: *The following python scripts were used to normalize the data and train the support vector machine (learning steps 1 to 3).*

Data Normalization

In [ ]:
```
def Normalize(X):
    '''
    The function transforms all variables to the interval [0,1]
    requires numpy as np
    '''
    X_norm=list()
    for column in X.T:
        column_norm=list()
        MAX=np.max(column)
        MIN=np.min(column)
        if MAX-MIN == 0:
            for entry in column:
                column_norm.append(0)
            X_norm.append(column_norm)
        else:
            for entry in column:
                column_norm.append((entry-MIN)/(MAX-MIN))
            X_norm.append(column_norm)
    X_norm=np.array(X_norm)
    return X_norm.T

X_norm=Normalize(X_modified)
print 'Number of patents/ observations ', len(X_norm)# X_modified.size
print 'Number of key-words/ variables = ', len(X_norm[0])
```

Learn from the data

Step 1 Split the dataset in two parts

In [ ]:
```
X_train, X_test, y_train, y_test = train_test_split(X_norm, y, test_size=0.15, random_state=0) # test_size regulates test data
```

Step 2 Set the parameters for tuning

In [ ]:
```
C_list=[10, 110, 210, 310, 410, 510, 610, 710, 810, 910, 1000]
Gamma_list= [1e-6,1e-5,5e-5,1e-4,5e-4,1e-3,5e-3,1e-2]
```

In [ ]:
```
tuned_parameters = [{'kernel': ['rbf'], 'gamma': Gamma_list,'C': C_list},
                    {'kernel': ['poly'], 'degree': [1,2,3],'coef0': [1,2,3], 'gamma': Gamma_list,'C': C_list},
                    {'kernel': ['sigmoid'],'coef0': [1,2,3], 'gamma': Gamma_list,'C': C_list}]
```

Step 3 Learning

In [ ]:
```
# desired cv parameters
cvs=[3,4]
# model evaluation parameters
in_scores = ['precision', 'recall','f1']

for input_cv in cvs:
    print 'Cross-validation parameter %s' % input_cv
    for input_score in in_scores:
        print("# Tuning hyper-parameters for %s" % input_score)
        clf = GridSearchCV(SVC(C=1,class_weight='auto'), tuned_parameters, cv=input_cv, scoring=input_score)
        clf.fit(X_train, y_train)
        print("Best parameters set found on development set:")
        print(clf.best_estimator_)
        print 'Number of support vectors - ', len(clf.best_estimator_.support_vectors_)
        print 'Indexes of support vectors - ', clf.best_estimator_.support_
        print 'The score function:',clf.scorer_
        print("Grid scores on development set:")
        for params, mean_score, scores in clf.grid_scores_:
            print("%0.3f (+/-%0.03f) for %r" % (mean_score, scores.std() / 2, params))
        print("Detailed classification report:")
        y_true, y_pred = y_test, clf.predict(X_test)
        print(classification_report(y_true, y_pred))
```

References

Ali, S. and Smith-Miles, K. A.: 2006, A Meta-Learning Approach to Automatic Kernel Selection for Support Vector Machines, *Neurocomputing* **70**(123), 173–186. Neural Networks Selected Papers from the 7th Brazilian Symposium on Neural Networks (SBRN 04), 7th Brazilian Symposium on Neural Networks.

Arora, S. K., Porter, A. L., Youtie, J. and Shapira, P.: 2013, Capturing new Developments in an Emerging Technology: An Updated Search Strategy for Identifying Nanotechnology Research Outputs, *Scientometrics* **95**, 351–370.

Arora, S. K., Youtie, J., Carley, S., Porter, A. L. and Shapira, P.: 2014, Measuring the Development of a Common Scientific Lexicon in Nanotechnology, *Journal of Nanoparticle Research* **16:2194**, 1–11.

Autor, D. H., Levy, F. and Murnane, R. J.: 2003, The Skill Content of Recent Technological Change: An Empirical Exploration, *The Quarterly Journal of Economics* **118**(4), 1279–1333.

Bassecoulard, E., Lelu, A. and Zitt, M.: 2007, Mapping Nanosciences by Citation Flows: A Preliminary Analysis, *Scientometrics* **70**, 859–880.

Boser, B., Guyon, I. and Vapnik, V. (eds): 1992, *A Training Algorithm for Optimal Margin Classifiers*, Proceedings of the Fifth Annual Workshop on Computational Learning Theory – COLT '92, p. 144.

Bresnahan, T. F.: 2010, General Purpose Technologies, *in* B. Hall and N. Rosenberg (eds), *Handbook of Economics of Innovation*, Vol. 2, Elsevier, pp. 763–791.

Burges, C. J. C.: 1998, A Tutorial on Support Vector Machines for Pattern Recognition, *Data Mining and Knowledge Discovery* **2**(2), 121–167.

Cortes, C. and Vapnik, V.: 1995, Support-Vector Networks, *Machine Learning* **20**(3), 273–297.

Duan, K.-B. and Keerthi, S.: 2005, Which Is the Best Multiclass SVM Method? An Empirical Study, *in* N. Oza, R. Polikar, J. Kittler and F. Roli (eds), *Multiple Classifier Systems*, Vol. 3541 of *Lecture Notes in Computer Science*, Springer, Berlin, Heidelberg, pp. 278–285.

Erdi, P., Makovi, K., Smomogyvári, Z., Strandburg, K., Tobochnik, J., Volf, P. and Zalángi, L.: 2013, Prediction of Emerging Technologies Based on Analysis of the US Patent Citation Network, *Scientometrics* **95**, 225–242.

Fischer, M., Scherngell, T. and Jansenberger, E.: 2009, Geographic Localisation of Knowledge Spillovers: Evidence from High-Tech Patent Citations in Europe, *Annals of Regional Science* **43**, 839–858.

Frey, C. B. and Osborne, M. A.: 2013, *The Future of Employment: How susceptible are jobs to computerization?*, Oxford University Programme on the Impacts of Future Technology.

Garfield, E.: 1967, Primordial Concepts, Citation Indexing and Historio-Bibliography, *Journal of Library History* **2**, 235–249.

Graetz, G. and Michaels, G.: 2015, Robots at work, *Center for Economic Peformance Discussion Paper* .

Griliches, Z.: 1990, Patent Statistics as Economic Indicators: A Survey, *Journal of Economic Literature* **28**, 1661–1707.

Guyon, I., Boser, B. and Vapnik, V.: 1993, Automatic Capacity Tuning of Very Large VC-dimension Classifiers, *Advances in Neural Information Processing Systems*, Morgan Kaufmann, pp. 147–155.

Halaweh, M.: 2013, Emerging Technology: What is it?, *Journal of Technology Management and Innovation* **8**(3), 108–115.

Hall, B. H., Jaffe, A. and Trajtenberg, M.: 2005, Market Value and Patent Citations, *RAND Journal of Economics* **36**(1), 16–38.

Hsu, C.-W., Chang, C.-C. and Lin, C.-J.: 2010, A Practical Guide to Support Vector Classification, *Technical Report*, Department of Computer Science and Information Engineering, National Taiwan University.

Jaffe, A., Trajtenberg, M. and Henderson, R.: 1993, Geographic Localization of Knowledge Spillovers as Evidenced by Patent Citations, *The Quarterly Journal of Economics* **108**(3), 577–598.

Kotsiantis, S. B.: 2007, Supervised Machine Learning: A Review of Classification Techniques, *Informatica* **31**, 249–268.

Lee, S., Yoon, B. and Park, Y.: 2009, An Approach to Discovering New Technology Opportunities: Keyword-Based Patent Map Approach, *Technovation* **29**, 481–497.

Lee, W. H.: 2008, How to Identify Emerging Research Fields Using Scientometrics: An Example in the Field of Information Security, *Scientometrics* **76**(3), 503–525.

Li, Y.-R., Wang, L.-H. and Hong, C.-F.: 2009, Extracting the Significant-Rare Keywords for Patent Analysis, *Expert Systems with Applications* **36**, 5200–5204.

Manning, C., Raghavan, P. and Schütze, H.: 2008, Introduction to Information Retrieval. online, accessed October 15 2014.
URL: *http://www-nlp.stanford.edu/IR-book/*

Mogoutov, A. and Kahane, B.: 2007, Data Search Strategy for Science and Technology Emergence: A Scalable and Evolutionary Query for Nanotechnology Tracking, *Research Policy* **36**, 893–903.

Nilsson, R., Björkegren, J. and Tegnér, J.: 2006, A Flexible Implementation for Support Vector Machines, *The Mathematica Journal, Wolfram Media, Inc.* **10**(1), 114–127.

Noyons, E., Buter, R., Raan, A., Schmoch, U., Heinze, T., S., H. and Rangnow, R.: 2003, Mapping Excellence in Science and Technology Across Europe. Part 2: Nanoscience and Nanotechnology. Draft Report EC-PPN CT2002-0001 to the European Commission. Leiden University Centre for Science and Technology Studies/Karlsruhe Fraunhofer Insitute for Systems and Innovations Research.

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. and Duchesnay, E.: 2011, Scikit-Learn: Machine Learning in Python, *Journal of Machine Learning Research* **12**, 2825–2830.

Pérez, F. and Granger, B. E.: 2007, IPython: a System for Interactive Scientific Computing, *Computing in Science and Engineering* **9**(3), 21–29.
URL: *http://ipython.org*

Porter, A., Youtie, J. and Shapira, P.: 2008, Nanotechnology Publications and Citations by Leading Countries and Blocs, *Journal of Nanoparticle Research* **10**, 981–986.

Ruffaldi, E., Sani, E. and Bergamasco, M.: 2010, Visualizing Perspectives and Trends in Robotics Based on Patent Mining, IEEE International Conference on Robotics and Automation, Anchorage, Alaska.

Srinivasan, R.: 2008, Sources, Characteristics and Effects of Emerging Technologies: Research Opportunities in Innovation, *Industrial Marketing Management* **37**, 633–640.

Stahl, B.: 2011, *What Does the Future Hold? A Critical View of Emerging Information and Communication Technologies and their Social Consequences*, Vol. 356 of *Researching the Future in Information Systems, IFIP Advances in Information and Communication Technology*, Springer, Berlin, Heidelberg.

Thompson, P.: 2006, Patent Citations and the Geography of Knowledge Spillovers: Evidence from Inventor- and Examiner-added Citations, *The Review of Economics and Statistics* **88**(2), 383–388.

Tseng, Y.-H., Lin, C.-J. and Lin, Y.-I.: 2007, Text Mining Techniques for Patent Analysis, *Information Processing and Management* **43**, 1216–1247.

Wolpert, D. and Macready, W.: 1997, No Free Lunch Theorems for Optimization, *IEEE Transactions on Evolutionary Computation* **1**(1), 67–82.

Yoon, B. and Park, Y.: 2004, A Text-Mining-Based Patent Network: Analytical Tool for High-Technology Trend, *Journal of High-Technology Management Research* **15**, 37–50.

Working Paper Series in Economics

recent issues

No. 71 *Florian Kreuchauff and Vladimir Korzinov:* A patent search strategy based on machine learning for the emerging field of service robotics, August 2015

No. 70 *Christian Feige:* Success rates in simplified public goods games - a theoretical model, June 2015

No. 69 *Markus Fels:* Mental accounting, access motives, and overinsurance, May 2015

No. 68 *Ingrid Ott and Susanne Soretz:* Green attitude and economic growth, May 2015

No. 67 *Nikolaus Schweizer and Nora Szech:* Revenues and welfare in auctions with information release, April 2015

No. 66 *Andranik Tangian:* Decision making in politics and economics: 6. Empirically constructing the German political spectrum, April 2015

No. 65 *Daniel Hoang and Martin Ruckes:* The effects of disclosure policy on risk management incentives and market entry, November 2014

No. 64 *Sebastian Gatzer, Daniel Hoang, Martin Ruckes:* Internal capital markets and diversified firms: Theory and practice, November 2014

No. 63 *Andrea Hammer:* Innovation of knowledge intensive service firms in urban areas, October 2014

No. 62 *Markus Höchstötter and Mher Safarian:* Stochastic technical analysis for decision making on the financial market, October 2014

No. 61 *Kay Mitusch, Gernot Liedtke, Laurent Guihery, David Bälz:* The structure of freight flows in Europe and its implications for EU railway freight policy, September 2014