Predicting high-quality pharmaceuticals patents using machine learning

Christian Rutzer and Matthias Niggli

December 2020

1 Introduction

In the interactive report Rutzer et al. (2020), we compare the quality of recent pharmaceutical innovations of countries, regions and firms (established ones and startups). Most of these innovations are protected and disclosed by patents. Thus, we can use a rich set of patent data to assess the expected quality of recently filed patents. Following the literature, we use the number of forward citations as a measure of patent quality (see, for example, Kogan et al. 2017, Hall et al. 2005). In particular, we predict whether a patent may become a top patent based on the number of expected forward citations over the next five years. In order to do this, we rely on machine learning algorithms. In this technical note, we describe in detail the training process and the performance of our machine learning algorithms. We structure this note as follows: First, we characterize the data used for training our models. Second, we explain the training process in more detail. Third, we show the prediction procedure for newly published patents. And finally, we explain how we combine different information sets to conduct meaningful analyses at the level of countries, firms or regions as shown in the report Rutzer et al. (2020).

2 Data

To train our models, we use patent data from the United States Patent Office (USPTO 2020).[1] In particular, we focus only on patents belonging to the technology field "16 pharmaceuticals" as defined by Schmoch (2008). Within this subset, we use patents filed between 2011 and 2015 (in both cases including the mentioned year) to train our models. The assumption behind this approach is that today's citation patterns for pharmaceutical patents and their determinants are well approximated by their 2011-2015 counterparts. Furthermore, we clean the dataset for patents that codify the same invention. In order to detect such equivalent patents, we exploit information from the OECD (2020*a*) and keep the patent with the earliest publication date among all of its equivalent USPTO and EPO patents. All in all, our final data contains a total of 25'337 patents.

Features

Since we ultimately want to make predictions about the quality of recently published patents, we can only consider information that is immediately available at the publishing date of a patent. Moreover, we don't take information into account which could cause a systematic distortion of predicted top patents between countries due to the number of top patents in the past. For example,

[1] We focus on patents from the USPTO to train our models, because citation rules differ among different patent offices. However, when it comes to predict the quality of newly published patents, we also include EPO patents. We will discuss later on why it should not be a major problem to use patents from different patent offices when it comes to predicting whether a patent will be a top patent.

we do not include information on the country of the inventor(s) or firm(s). In total, we have 47 such features to train our prediction models. These variables can be divided into three groups.

The first group of variables contains direct information about a patent, such as the number of claims, the number of cited patents and the number of cited non-patent literature. All variables belonging to this group are listed in Table 7 in the Appendix at the end of the document.

The second group contains information about the dynamics of the technologies to which a patent is assigned to. Table 8 in the Appendix lists all variables capturing technology dynamics. To derive these, we exploit the following information: Each patent belongs to at least one International Patent Classification (IPC). For each 4-digit IPC group, we can thus create several variables, such as the growth rate of the number of patents within an IPC group or the number of patents per IPC and year. To derive these variables, we use all distinct (ie, non-equivalent) USPTO and EPO patents of all technology fields. We use the patent's priority year to assign information about technology dynamics. If a patent is assigned to several IPCs, we take the simple average of the values for each IPC-group of the respective technology variable.

Finally, the third group of variables contains information about a patent's inventor(s) and application firm(s). Examples are the number of patent applications of a firm in the last five years, the number of patent applications of a firm within the same IPC-group in the last five years or the average age of a firm's patent portfolio. All variables of this group are listed in Table 9 of the Appendix.

Outcome variable

Our aim is to predict the quality of patents. To proxy quality, we use the number of forward citations a patent receives within five years after publication. Therefore, to train our models, we use only patents that have been published at least five years ago. For these patents, we first calculate the number of forward citations within the first five years after publication. Next, we split the patents into three groups based on their number of received forward citations. Patents with forward citations below the 50% decile are assigned to the first group p_l, between the 50% and 90% decile to the second group p_m and with a number equal to or lager than the 90% decile to the third group p_h. In the following, we call the latter group top patents.

There is one particular challenge in using this approach. The general citation intensity of patents can vary over time. In order to take this into account, we calculate the citation deciles on a yearly basis. In doing so, we use the patent's publication year as the reference point for time.[2] Table 1 shows the cut-off values of citations used to assign patents to the three different groups depending on the respective publication year.

Table 1: Citation cut-offs for classifying the patents

Publication year	cut-off p_m	cut-off p_h
2011	6	23
2012	5	24
2013	4	21
2014	4	19
2015	2	14

In a next step, we divide the data into a training, validation and test set. We use 65% (16'470 patents) of the data for training, 15% (3'800 patents) for validation and the remaining 20% (5'067

[2] We use a different time indicator for the outcome variable and the feature variables. For the feature variables, we use the priority year and not the publication year to allocate a patent to a particular time point. This is because the feature variables aim to capture the knowledge of inventors and firms. And since the priority year of a patent comes closest to the actual time of an invention, it seems best suited to capture the knowledge of the parties involved at this point in time. However, since patents can only be cited after publication, using the publication year seems best suited to measure the number of citations.

patents) for testing. To construct each data set as equal as possible, we draw stratified samples using the patent quality classes as sub-populations. This means each of the three sets contains 10% p_h-patents, 40% p_m-patents and 50% p_l-patents. As a last step, we normalize the input variables. This is necessary for some classes of machine learning algorithms, such as Neural Networks. Otherwise, features with a higher value will gain greater importance by construction. Since some of our included input variables can change over time (e.g. the number of backward citations may not be time-invariant), we subtract from each variable the mean of the respective year and divide it by the standard deviation. Moreover, in order to avoid information leakage from the training or validation set to the test set, we normalize the test data separately. After these steps, we are ready to train our models.

3 Training and evaluating different classification models

In the following, we describe the training process of our models. We train several classification models: Two Neural Networks (NN_1 and NN_2) and two Random Forests (RF_1 and RF_2).[3]

The ultimate objective of our analysis is to compare the quality of newly published pharmaceutical patents of different countries, regions and firms. After classifying individual patents, we will thus aggregate them and calculate the shares of top patents by countries, regions and firms (i.e. we calculate the sum of all recently published patents that are predicted to be top patents and divide it by the total number of recently published patents). In general, these shares can deviate from the "true" value either due to false negatives (which would, ceteris paribus, underestimate the relative number of top patents) or false positives (which would, ceteris paribus, overestimate the relative number of top patents) from our classification model. Since it is not clear which type of error is worse in our framework, we use the F1-score with respect to top patents as a measure to compare the goodness of prediction of our four classification models.[4]

It is important to note that our data is unbalanced. If this is not addressed properly, top patents may receive too little weight in the training process. As a result, our models could not perform well in finding top patents (ie, we would have too many false negatives). At the same time, putting more weight on classifying top patents correctly could result in more false positives, ie, more patents that are falsely classified by the algorithm as being a top patent. Therefore, we train each of our class of models (ie, Random Forst and Neural Network) by explicitly considering the unbalanced data in one case and not in another.

Random Forest Let us start with the Random Forest. For RF_1, we consider all data for training. For RF_2, we undersample patents belonging to the class p_m and p_l in such a way to have exactly the same number of patents as of the minority class p_h.[5]

We use a grid search in order to determine several hyperparameters. In particular, we tune the number of trees, the number of randomly selected variables at each split and the number of minimal observations a node must have in order to stop the splitting process. We then select the group of hyperparameters leading to the highest F1-score for top patents when applied to the validation set. Table 2 shows our optimal set of hyperparameters for both Random Forest models.

[3] We have also tried stacked ensemble models using the predicted probabilities of our models as inputs. In particular, we used a simple Multinominal Logit model, a Random Forest and a Boosting algorithm. However, the trained ensembles have not improved the performance significantly. Therefore, we did not further pursue this approach.

[4] The F1-score with respect to top patents is calculated as the harmonic mean between Recall (ie, what is the share of true top patents that the algorithm detects) and Precision (ie, what share of top patents classified by the algorithm are true top patents). Therefore, it reflects both types of errors.

[5] We also generated synthetic data in order to provide additional training data for the p_h class. However, this approach did not really improve the performance of our models, so we refrained from doing so.

Table 2: Hyperparameters of the Random Forests

Hyperparameter	Set of possible values	Optimal value RF_1	Optimal value RF_2
# of trees	{100, 125, ..., 400}	375	180
# of randomly selected variables	{2, 4, ..., 10}	6	2
# of observations at a terminal node	{1, 4, ..., 13}	13	10

Neural Network Next, we describe the training of our Neural Networks. We train two simple feed-forward networks. Each network contains a single hidden layer. This seems sufficient due to our rather simple structured data.[6] Our first Neural Network NN_1 gives every observation the same weight. Our second Neural Network NN_2 gives top patents (ie, patents of the group p_h) twice the weight in the loss function.[7] The latter takes the imbalanced nature of our data into account. Again, we use a grid search to find optimal values for several hyperparameters based on the F1-score on the validation data. Since it takes much longer to train the Neural Networks and since we have much more hyperparameters to tune than for the Random Forests, we do a random grid search. This means we randomly select only 80% of all of our parameter tuples. Table 3 lists the optimal parameter values for each of our tuned hyperparameters.

Table 3: Hyperparameters of the Neural Networks

Hyperparameters (non-tuned)	Set of possible values	NN_1	NN_2
Loss function	Cat. crossentropy	Cat. crossentropy	Cat. crossentropy
Hyperparameters (tuned)			
Optimizer	{Adam, RMSProp, SGD}	Adam	Adam
Nodes Input-Layer	{4, 16, 24, 32, 42}	42	24
Nodes Hidden-Layer	{4, 16, 24, 32, 42}	16	32
Dropout Input-Layer	{0.05, 0.1}	0.1	0.05
Dropout Hidden-Layer	{0.05, 0.1}	0.05	0.05
L2 regularization Input-Layer	{0.001, 0.01, 0.1}	0.001	0.001
L2 regularization Hidden-Layer	{0.001, 0.01, 0.1}	0.001	0.1
Learning rate annealing	{0.05, 0.1}	0.1	0.1

Model performance In the following, we use our test data to compare the goodness of prediction of the previously trained and tuned models. We only show the performance of predicting top patents correctly, because this is what we are interested in.[8] Table 4 shows the results. To interpret the results, consider the second row. A Recall of 0.57 means that the RF_1 model is able to detect 57% of the top patents in the test data. A Precision of 0.78 means that 78% of the patents of the test data classified by the RF_1 model as top patents are "true" top patents. Finally, the $F1$-score is the harmonic mean between Recall and Precision. For two values of Recall and Precision that have the same simple average, the harmonic mean is larger the closer the values of Recall and Precision are to each other.

[6] We have also experimented using more hidden layers. However, adding more layers has not really improved the performance. Thus, we stick to a rather simple network architecture.

[7] Since Neural Networks tend to require more observations for training than Random Forests, we have not used undersampling here to take the unbalanced nature of our data into account.

[8] Although we are only interested in detecting top patents and non-top patents, we used three classes for training. The reason for this is that the goodness of prediction of top patents is better when considering models with three classes than with only two classes. This is probably due to the fact that p_m patents have a systematically different structure than p_l and p_h patents. Thus, if the p_m class is not explicitly implemented, then the algorithms might wrongly classify a larger number of otherwise as p_m classified patents as p_h ones. Thus, we train our algorithms with three classes and only mix the patents p_m and p_l together to the class of non-top patents for our final analyses.

Table 4: Goodness of prediction of top patents

Model	Recall	Precision	F1
RF_1	0.57	0.78	0.65
RF_2	0.63	0.71	0.67
NN_1	0.58	0.71	0.64
NN_2	0.78	0.57	0.66

As it is also apparent in the table, both models trained to explicitly taking the unbalanced nature of our data into account (RF_2 and NN_2) have a higher Recall and a lower Precision relative to the models (RF_1 and NN_1, respectively) that treats all observations with similar importance. This makes intuitively sense. Both models, RF_2 and NN_2, have been trained to put more emphasis on identifying top patents. Therefore, not surprisingly, they are able to identify a greater proportion of the true top patents in the test data. But this comes at the expense of a lower Precision, ie, these models wrongly classify a larger number of non-top patents as top patents.

In order to put the results into perspective, we compare them with the literature on patent quality classification. To our knowledge, there are two other papers that also predict the quality of patents based on the number of forwards citations using machine learning. Lee et al. (2018) use pharmaceutical patents from the USPTO published between 2000 to 2009. In one of their setups, they use the number of forward citations after five years of publication as a proxy for patent quality (additionally, they consider 3 and 10 years). In contrast to our approach, they assign patents to 4 instead of 3 classes and use absolute numbers of citations as class boundaries, a circumstance that could be problematic due to the change of the citation distribution over time. Furthermore, they only use 22 input variables and do not perform hyperparameter tuning (at least this is not explicitly mentioned). This shortcomings maybe the reason for the rather disappointing performance in detecting top patents. The Recall regarding the group of top patents was only 0.17 and the Precision 0.56.

Chung & Sohn (2020) train a model to predict top patents of the semiconductor industry using patents from the USPTO published between 2000 and 2015. They assign patents to three groups based on the number of forward citations on yearly deciles (0-0.4, 0.4-0.8, 0.8-1). In contrast to our approach, they also consider the text of a patent as further source of input data. With this additional information, they achieve to significantly improve the classification performance. In particular, they report a Precision regarding the group of top patents of 0.78, a Recall of 0.75 and F1-score of 0.77. Besides the somewhat larger group of top patents (20% instead of 10%) and a different technological field (ie, semiconductors), the inclusion of patent texts might play an important role for this better performance. A clear indication in that direction is given by their reported performance values if patent texts are not considered. In this case, the performance is in line with our results, namely a Precision of 0.70, a Recall of 0.67 and an F1 of 0.68. Therefore, additionally taking text data into account would certainly be a rewarding extension for the future.

4 Predicting the quality of new patents

In a next step, we use the trained models to evaluate the quality of newly published pharmaceutical patents.[9] In doing so, we consider all pharmaceutical patents published by the USPTO and EPO for which the first publication date is 2017 or later (ie, the earliest publication date of any equivalent patent is in 2017 or later).[10] Our data includes patents published until April 2020. In total, we assess the quality for 8'811 pharmaceutical patents.

[9] In order to find pharmaceutical patents, we use the technology classifications of Schmoch (2008).

[10] In contrast to the training step, we now also include EPO patents. This increases the number of innovations we are being able to take into account for our analysis, especially because the EPO publishes also patent applications and not only granted patents.

Preparing the data For a newly disclosed patent it is not yet clear how many forward citations it will receive in the next five years after publication, and, thus whether it could be a top patent. But it is possible to derive all variables that we have previously used to train our algorithms. As for the training data, we again normalize each variable by subtracting the corresponding mean and dividing it by its standard deviation. Usually, one uses the moments of the training data to normalize the variables of new observations. However, as previously mentioned, some input variables are unlikely to be time-invariant. And since we have a whole cohort of new data at our disposal, we do the normalization by using the mean and standard deviation of this new cohort. This does not pose a problem since we are only interested in identifying the 10% most cited patents and not in predicting the class probabilities as such. Specifically, we use a variable's mean and standard deviation of either the 2017-2018 or 2019-2020 period, depending on the patent's publication date. In addition, we normalize EPO and USPTO patents separately to take into account possible systematic differences. An example for such systematic bias might be the number of backward citations due to diverging citation rules.[11] After having normalized the input variables, we can use them in our previously trained algorithms to make statements about the quality of newly published patents.

Results of predicting the class probabilities For each patent we get a probability of belonging to one of the classes p_l, p_m or p_h. Table 6 illustrates these class probabilities based on the NN_1 model for some randomly selected new patents. We show only the results of one model, because the class probabilities predicted by the Random Forest algorithms are distorted probabilities.[12] Thus, it is not possible to directly compare the predicted probabilities of different models.

Table 5: Example of predictions for some newly published patents

Patent number	Model	p_l	p_m	p_h
US10493084	NN_1	0.01	0.15	0.84
US10342786	NN_1	0.01	0.16	0.83
EP3488851	NN_1	0.02	0.09	0.89
EP3513778	NN_1	0.14	0.51	0.35
EP3513785	NN_1	0.85	0.15	0.00

For our analysis, however, this is not problematic. Our definition of top patents is based on a relative ranking. We sort all patents in descending order based on the predicted probability of belonging to the p_h class. Afterwards, we classify the first 10% as top patents. We do this separately for each algorithm. Table 6 shows the degree of agreement among our algorithms.

To interpret the results, consider the first cell of the second column. It states that 77% of the patents classified as top patents by the RF_1 model are also classified as top patents by the RF_2 model. For non-top patents the degree of agreement between both algorithms is 98%. The other pairwise comparisons show similarly high agreements–an indication that our predictions are reasonably robust.

[11] Different citation rules are the main reason why we solely use USPTO patents to train our classification algorithms. For out-of-sample predictions of the class-probability, however, this should be, if at all, only a minor issue for at least two reasons. First, in contrast to the outcome variable, which is our most important variable when training our models, only a few of our input variables may have systematically different values. Second, we normalize the input variables of EPO and USPTO patents separately.

[12] This is due to how probabilities are determined in a Random Forest. First, each trained tree is used to assign a patent to a class. The class probability of the whole Random Forest is then received by taking the average of all trees and for each particular class.

Table 6: Degree of agreement among our algorithms

	RF_2	NN_1	NN_2
RF_1 Top patents	0.77	0.69	0.73
RF_1 Non-top patents	0.98	0.96	0.97
RF_2 Top patents	—	0.66	0.71
RF_2 Non-top patents	—	0.96	0.97
NN_1 Top patents	—	—	0.81
NN_1 Non-top patents	—	—	0.98

5 Creating the final output

This leaves us to describing how we derive the quality of recent patents of countries, regions and firms used in the report of Rutzer et al. (2020).

Let's start with the patent quality of **countries and regions**. In order to assign patents to geographic entities, we use the residence of the inventors.[13] If, for example, two inventors from Switzerland and one from Germany were involved in the development of a patent, we assign such a patent to 2/3 to Switzerland and to 1/3 to Germany. For the assignment to regions we use the region classification of the OECD (2020*b*). Our EPO patents obtained from the OECD already contain this information. For the USPTO patents, we derive the regions based on address information of inventors.

We also analyze patents of the **15 largest pharmaceutical firms**. The selection of these companies is based on the number of patents published between 2010 and 2015. Since the name of a company may differ between different patents (or even contain spelling errors), we first make some data cleaning in order to catch as many patents as possible from a given firm. First, we capitalize all words and remove unmeaningful words such as "company" or "corporation" (we also consider unmeaningful words of several other languages than English). Afterwards, we perform string matching between the 15 most frequently mentioned company names (ie, we consider the 15 largest pharmaceutical firms in terms of patents) and all other firm names among pharmaceutical patents. We match firm names based on the Levenshtein distance. This allows us to detect and match different spellings of a firm name, such as "Böhringer Ingelheim" (ie, the most frequent occurrence in the patent data) and "Boehringer Ingelheim". We also perform a manual search by simply using some key words of the 15 most important pharmaceutical companies. An example is a search using the word "Roche", which captures, among others, Roche's subsidiary Roche Glycart. However, we focus only on patents of the core company. For example, patents by Genentech are not attributed to Roche. Finally, we analyze patents of **pharmaceutical startups**. We define a company as a pharmaceutical startup if its very first pharmaceutical patent was registered no earlier than in 2010. In order to omit less relevant startups, we consider among this subgroup only those firms that conduct R&D by themselves and have had at least 6 pharmaceutical patents published from 2010 onward.

After having defined important pharmaceutical firms and startups and having attributed patents to countries, and regions, we add information on whether a patent maybe a top patent or not. This allows us to calculate the number of top patents among all newly published pharma patents at the level of countries, regions, firms and startups. The results are presented in the interactive report of Rutzer et al. (2020).

[13] In the case of Switzerland, it should be noted that we assign patents invented by cross-border commuters to Switzerland. This is important, as otherwise Switzerland's patent output could be significantly underestimated. For details on the methodology and the size of the underestimation, see Niggli et al. (2020).

Computational Details

All calculations are done with **R** (version 4.0.3). For the preparation of the data we used mainly the packages **dplyr** and **data.table**. For the training of the Neural Networks, we used **R-Keras** and for the remaining models **mlr3**. Most calculations were performed at **sciCORE** scientific computing center at University of Basel.

References

Chung, P. & Sohn, S. Y. (2020), 'Early detection of valuable patents using a deep learning model: Case of semiconductor industry', *Technological Forecasting and Social Change* **158**, 1201–1210.

Hall, B. H., Jaffe, A. & Trajtenberg, M. (2005), 'Market value and patent citations', *The RAND Journal of Economics* **36**(1), 16–38.

Kogan, L., Papanikolaou, D., Seru, A. & Stoffman, N. (2017), 'Technological innovation, resource allocation, and growth', *The Quarterly Journal of Economics* **132**(2), 665–712.

Lee, C., Kwon, O., Kim, M. & Kwon, D. (2018), 'Early identification of emerging technologies: A machine learning approach using multiple patent indicators', *Technological Forecasting and Social Change* **127**, 291–303.

Niggli, M., Rutzer, C. & Filimonovic, D. (2020), Cross-border commuting and inventions "made in switzerland", Interactive report. The report can be accessed at: `https://innoscape.ch/en/publications/cross-border-commuting-and-inventions-made-in-switzerland`.

OECD (2020*a*), Sti microdata lab, Raw patent data. Url: `https://survey2018.oecd.org/Survey.aspx?s=85071fe20881410ebed2258aa9f93561`.

OECD (2020*b*), Territorial grids, Oecd report. Url: `http://www.oecd.org/cfe/regionaldevelopment/territorial-grid.pdf`.

Rutzer, C., Niggli, M. & Filimonovic, D. (2020), What's next? swiss pharma in the international innovation race, Interactive report. The report can be accessed at: `https://innoscape.ch/en/publications/swiss-pharma-international-innovation-race`.

Schmoch, U. (2008), Concept of a technology classification for country comparisons, Report, World Intellectual Property Organisation (WIPO).

USPTO (2020), Patentsview, Patent data api. Url: `https://www.patentsview.org/web/#viz/comparisons`.

A List of variables used to train the algorithms

Table 7: Variables directly related to a patent

Variable	Source	Description
Num_firm	Own calculations based on OECD and USPTO data	Number of assignees
Num_inv	—”—	Number of inventors
Num_ctry_firm	—”—	Number of countries the firms are from
Num_ctry_inv	—”—	Number of countries the inventors are from
Fam_size	—”—	Number of patents within a patent family
Back_cits	—”—	Number of cited patents
Claims	—”—	Number of claims
Pat_scope	—”—	Number of different 4-digit IPC classes
Originality	—”—	Herfindahl Index of backward citations i.e., how dispersed are the backward citations of a patent across different IPCs
Radicalness	—”—	Sum across all backward cited patents of the number of different IPCs a backward cited patent belongs to which are not part of the citing patent divided by the total IPCs of the backward cited patent
Tri_pat_fam	—”—	Indicates whether the patent belongs to a triadic patent family (ie, application at the USPTO, the EPO and the JPO)
Npl_cits	—”—	Number of cited non-patent literature
Mean_age_back_cits	—”—	Mean age of backward citations
Std_age_back_cits	—”—	Standard deviation of the age of backward citations
Mean_age_npl	—”—	Mean age of cited non-patent literature
Std_age_npl	—”—	Standard deviation of the age of cited non-patent literature

Table 8: Variables capturing technology dynamics

Variable	Source	Description
Num_per_ipc	Own calculations based on OECD and USPTO data	Number of patents per IPC and year
Ipc_gr	—"—	Yearly growth of Num_per_ipc
Ipc_share	—"—	Number of patents per IPC relative to total number of patents
Ipc_share_gr	—"—	Yearly growth of Ipc_share
Ipc_abs_gr	—"—	Absolute change in the number of patents per IPC in the last five years
Ipc_ave_gr	—"—	Average growth rate of the number of patents per IPC in the last five years

Table 9: Variables capturing information about the inventor(s) and assignee(s) (which are mainly firms)

Variable	Source	Description
Firm_know_age	Own calculations based on OECD and USPTO data	Average age of patents of a firm (last five years)
Inv_know_age	—"—	Average age of patents of an inventor (last five years)
Firm_diff_ipc	—"—	Number of different IPCs a firm has patents in (last five years)
Inv_diff_ipc	—"—	Number of different IPCs inventors have patents in (last five years)
Firm_num_sim_ipc	—"—	Number of patents a firm has in the same technology as the patent (last five years)
Inv_num_sim_ipc	—"—	Number of patents inventors have in the same technology as the patent (last five years)
Firm_num_pats	—"—	Number of total patents of a firm (last five years)
Inv_num_pats	—"—	Number of total patents of inventors (last five years)
Firm_ipc_index	—"—	Herfindahl index of the share of patents belonging to the same IPC of firms
Inv_ipc_index	—"—	Herfindahl index of the share of patents belonging to the same IPC of inventors
Firm_know_age_ipc		Firm's average age of patents having the same IPC as the patent (last five years)
Inv_know_age_ipc	—"—	Inventor's average age of patents having the same IPC as the patent (last five years)
Rad_five_firm	—"—	Average radicalness of patent portfolio of a firm (last five years)
Rad_five_inv	—"—	Average radicalness of patent portfolio of an inventor (last five years)
Orig_five_firm	—"—	Average originality of patent portfolio of an inventor (last five years)
Orig_five_inv	—"—	Average originality of patent portfolio of an inventor (last five years)
Bwd_cits_five_firm	—"—	Average backward citations of patent portfolio of a firm (last five years)
Bwd_cits_five_inv	—"—	Average backward citations of patent portfolio of an inventor (last five years)
Npl_cits_five_firm	—"—	Average number of non-patent literature cited by a firm's patent portfolio (last five years)
Npl_cits_five_inv	—"—	Average number of non-patent literature cited by an inventor's patent portfolio (last five years)

Claims_five_firm	—”—	Average number of claims of a firm’s patent portfolio (last five years)
Claims_five_inv	—”—	Average number of claims of an inventor’s patent portfolio (last five years)
Tri_pat_five_firm	—”—	Average number of triadic patents in a firm’s patent portfolio (last five years)
Tri_pat_five_inv	—”—	Average number of triadic patents in an inventor’s patent portfolio (last five years)
Patent_scope_five_firm	—”—	Average number of distinct 4-digit IPC classes to which the patents of a firm belongs to (last five years)
Patent_scope_five_inv	—”—	Average number of distinct 4-digit IPC classes to which the patents of an inventor belongs to (last five years)
Uni	—”—	Indicator whether assignee(s) is (are) an academic institution; a private firm or both