

# How to play the “Names Game”: Patent retrieval comparing different heuristics

Julio Raffo<sup>1,2,\*</sup>, Stéphane Lhuillery<sup>1</sup>

<sup>1</sup> EPFL College du Management de la Technologie  
Station 5 – Odyssea 1.18  
CH-1015 Lausanne  
stephane.lhuillery@epfl.ch

<sup>2</sup> CEPN – Université Paris Nord  
99, av Jean-Baptiste Clément  
F-93430 Villetaneuse  
raffo@seg.univ-paris13.fr

\* Corresponding author

**Comments are welcome**

---

Abstract: We propose to apply and assess different algorithms in order to provide automatic retrieval of inventors in patent datasets. We investigate the different solutions matching the 2006 EPO dataset with an inventor’s list from the EPFL. Our results suggest that the 2-gram method is the best matching algorithm to use. It has to be combined with a multiple parsing method and a disambiguation procedure including multiple filters. In this way, compared to a simple string match process, the overall recall rate increases by 10% when the precision rate is kept constant while the precision rate increases by 12% when the recall rate is kept constant. This suggests that a useful tradeoff is to be made and that the usual focus on precision rates is specious. The results are supported by econometric results exploring the determinants of EPFL patenting.

---

Keywords: patents, inventors, firms, university, names, matching algorithm.

JEL: C63, C81, C88, O34.

## **Table of contents**

1. Introduction.....	3
2. Matching algorithms .....	5
2.1. Existing string matching algorithms .....	5
2.2. On algorithms' performances .....	6
3. Data sources and benchmark set .....	7
4. The matching stage .....	8
5. The parsing stage .....	11
6. The filtering stage .....	12
7. Conclusion .....	14
8. References.....	18
Appendix.....	20

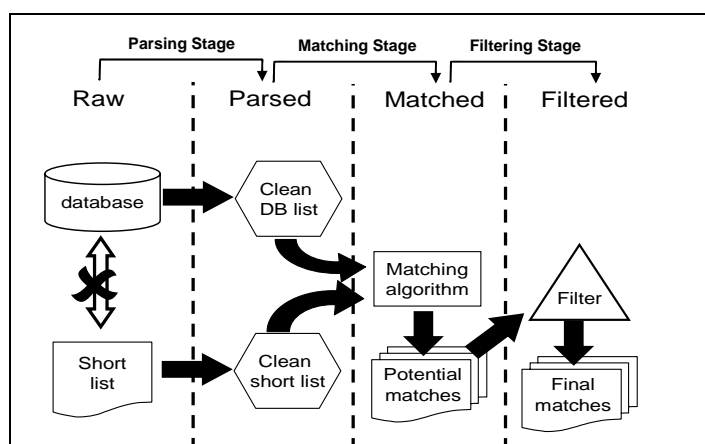
Acknowledgement: The EPFL TTO and the EPFL Human Resources are acknowledged for data availability. When producing this paper, we also benefited considerably from the expertise of several computer scientists and engineers. We specially want to acknowledge the collaboration of Alexandre Gonçalves (UFSC), Lautaro Matas (UBA/CAICYT) and Fernando Lladós (ITBA/UPM). The latter was of great help when implementing the N-grams algorithms and he is primarily responsible for the development of its highly performing indexation. We also acknowledge D522 Guellec, J520 Rollinson, G630 Thomas for their comments and the participants at the EPO Conference on Patent Statistics for Policy Decision Making, Venice International University, San Servolo, Venice, Italy, 1-3 October 2007. Any errors are our own.

## 1. Introduction

The provision to economists of large data sets on publications (patents or scientific articles) and of important capacities of calculation is recent. These facts induce researchers to consider two main methodological strategies: the first is to match words contained in abstracts either to validate ownership (e.g. Cassiman *et al.*, 2007) or to match documents (e.g. Lissoni and Montobbio, 2006). The second path – a more usual and restricted one – is trying to match names between a large patent or scientific publications’ dataset and an external list of firms (Bound *et al.*, 1984; Derwent, 2002; Mageman *et al.*, 2006; Hall, 2007; Kerr and Fu, 2007; Agarwal *et al.*, 2007;) or personal names (Fleming *et al.*, 2004; Singh, 2005; Jones, 2005; Trajtenberg *et al.*, 2006; Kim *et al.*, 2006; Hoisl, 2007; Thursby *et al.*, 2007; Mariani *et al.*, 2007; Azoulay *et al.*, 2007; Marx *et al.* 2007).

This latter “names game” has become a common activity for researchers working on patents. Any name matching procedure – such as those used in the articles mentioned above – can be conceptualized as three sequential stages, as depicted in Figure 1. The first stage includes all the rather simple tasks applied to the raw data – both publications’ database and researchers or firm names list – aiming to clean any noise such as different cases, corrupted characters, double spaces, *etc.* The second stage consists in applying any given matching algorithm to obtain a list of potential matched pairs. In the third and final stage, these matches can be filtered by using any complementary information about each pair of potential matches in order to disambiguate true matches from false ones. These three stages can be entitled the parsing stage, matching stage and filtering stage respectively. We will refer to them in this way hereafter.

*Figure 1 – The three stages of name matching process*



There are two major difficulties in this game: the first is the choice of steps or procedures to apply within each stage. The second concerns the procedures sequence: the choice made for each step determines the problem to solve and consequently the performances of procedures applied in the following stages. The identification of the best procedures sequence is therefore not that straightforward in a problem where there is no silver bullet solution.

The available literature usually implements only one procedure sequence<sup>1</sup> and not necessarily all of the three stages. The relative reliability of techniques used in the different papers is thus still an open question. Since the matching process is a time consuming but raising activity for researchers, the identification of the best heuristics can be a source of productivity and comparability of their results.

The present methodological paper proposes evaluating several possible heuristics, comparing possible alternatives inside the three stages and the interactions between these stages. Using a large dataset on patents (the European Patent Office (EPO) Worldwide Patent Statistical Database called “PATSTAT” thereafter), we identify patents for a set of inventors from the Ecole Polytechnique Fédérale de Lausanne (EPFL thereafter). The EPFL Technology Transfer Office (TTO) provided a list of 349 inventors listed in 1995-2005 EPFL patents. The TTO information on these inventors is completed with the data from the EPFL human resources. The combination of the two sources allows us to build a small but precise benchmark set of EPFL inventors and their patents (1,830 pairs). The EPFL benchmark data set is then used in order to assess the performances of different heuristics to be applied by future research.

Our results suggest that all three stages have to be applied in order to decrease the bias of the matched sample with respect to the real population. At the matching stage, the weighted 2-gram is proof of the best algorithm as it gives a good trade-off between recall and precision. However, it has to be combined with a previous multiple parsing and a subsequent disambiguation procedure implementing alternatively multiple filters. By doing this an efficiency gain can be achieved in both terms of recovering true matches and discarding incorrect ones. Obviously some errors will still remain. We acknowledge that a subsequent manual check – including the active strategy of contacting the firms or researchers by email or phone – can boost the precision rate in a great extent. However, some pairs will always be difficult to identify accurately.

The paper is organized as follows. Section 2 introduces the main types of algorithms available and their performances. Section 3 briefly presents the data used including the EPFL benchmark data set. The complexity of the problem and the wide number of possible combinations among the different steps from the different stages is a challenge from a didactic point of view. We propose here to focus first on matching algorithms and their performances applied on data sets already fully parsed (Section 4). Once the reader is aware of the properties of the different matching algorithms, we propose in Section 5 to come backward on the parsing stage in order to assess associated problems and the solutions to implement. Similarly, section 6 explores the impact of the different filters applied to the patent matches coming from the best matching algorithms identified in section 4. A final section concludes.

---

<sup>1</sup> After Trajtenberg et al. (2006), the paper of Marx et al. (2007) or Agarwal et al. (2007) are examples of papers providing their matching strategies.

## 2. Matching algorithms

### 2.1. Existing string matching algorithms

The choice of the matching algorithm is critical as it manages two usual problems in the matching process: when a pair of persons (or firms) are considered as different when in fact they are not (called type I errors, or false negative); when a pair of persons or firms are considered as the same ones when they are not (type II errors, or false positive). When *Type I* error occurs, it decreases the *Recall* rate<sup>2</sup> whereas *Type II* error decreases the *Precision* rate<sup>3</sup>. When applying a matching algorithm we will, theoretically, increase the recalled matches to a given name which means a decrease of *Type I* error. But, while the use of these techniques will enlarge the recall of misspelled or differently articulated names, it will also enlarge the incorrectly imputed matches, resulting in an increase of *Type II* error.

Beyond simple text string matching often used in economics of science and innovation (e.g. chosen by Fleming et al., 2004; Singh J., 2005; Kerr & Fu, 2007; Marx et al., 2007), matching algorithms can be chosen among three main families: Phonetic algorithms, Edit distance algorithms and vectorial decomposition algorithms.

*Phonetic algorithms* regroup phonemes by sound proximity using a simple set of rules. The best known are the Soundex algorithm, originally developed by Russell and O'Dell in 1918 and applied for instance by Trajtenberg *et al.* (2006) and Kim *et al.* (2006), and the Metaphone algorithm developed by Lawrence Philips. Many alternative phonetic based algorithms have been developed including variations of these two, such as the Daitch-Mokotoff Soundex, NYSIIS, Double Metaphone, Caverphone, Caverphone 2.0, Phonix, Onca, Fuzzy Soundex, etc. As an example, let's consider the string 'Aebischer Patrick'. Using the 6 phonetic rules from Soundex this string is recoded as A126, while using the 16 phonetic rules from Metaphone it is recoded as EBSXRPTRK. The results are exactly the same for both algorithms for the similar string 'Abisher Patric'.

*Edit distance algorithms* are also based on a simple precept, which is that any text string can be transformed into another one by applying a given number of plain operations. The Levenshtein distance between two patterns A and B is defined as the minimum number of operations such as changes, insertions and deletions required to change B to A (Levenshtein, 1966). Transforming 'Abischer Patric' into 'Aebischer Patrick' requires for example only 2 insertions.

Finally, the family of *vectorial decomposition algorithms* is basically a comparison of the elements of both strings. The elements of a given text string can be defined in different ways. The N-gram algorithm decomposes the text string into elements of N characters, called grams, on a moving windows basis. For example, a 3-gram decomposition of *Aebischer Patrick* will include 15 3-grams: AEB, EBI, BIS, ISC, SCH, CHE, HER, ER\_, R\_P, \_PA, PAT, ATR, TRI, RIC and ICK. When compared with the name *Abischer Patric* for example, this pair shares only nine trigrams: PAT, ATR, TRI, RIC, BIS, ISC, SCH, CHE and HER. The Token algorithm splits the text string by its blank spaces into different elements, called tokens. In our example, 'Aebischer' and 'Patrick' are the only two tokens identified. Once both compared text strings

---

<sup>2</sup> Recall rate=CR/(CR+CM) where, CR is Correct Recall, CM is Correct Missing (Error type I or false negative).

<sup>3</sup> Precision Rate=CR/(CR+IR) where, CR is Correct Recall, IR is Incorrect Recall (Errors type II or false positive).

are decomposed into elements, a similarity indicator can be computed by applying the cosine distance between both vectors of elements (either grams or tokens) or any other measure.

## 2.2. *On algorithms' performances*

As suggested in the above examples, each algorithm has its own merits. Phonetic based algorithms are more efficient at managing alike sounds based on misspellings. Also pre-calculation and indexation are possible here, which implies that after calculating the phonetic code for each name on a list there is no need for recalculate it again. It reduces computational costs. Conversely, phonetic algorithms are not convenient for permutation of characters and even less so for permutation of names. In our example, the string 'Patrick Aebischer' gives P362 in Soundex and PTRSBSXR in Metaphone, to be respectively compared with A126 and EBSXRPTRK found for 'Aebischer Patrick' with the same algorithms? Furthermore, as they rely on a perfect string match, they do not allow the computation of a degree of matching different from 0 (false) or 1 (true). The Edit distance algorithm family offers a more precise matching measure and manages typing or spelling errors effectively but is not well suited to long permutations, for example swapping between the first and last names. Converting our example "*Aebischer Patrick*" into "*Patrick Aebischer*" requires 16 operations (8 deletions and 8 insertions). It also requires a longer computational time and, even worse makes reduction by pre-calculation or indexation impossible. More attractive are the N-gram algorithms which work effectively on misspellings as well as large string permutations. N-grams also allows pre-calculation and indexation, lowering the computational costs considerably, these costs certainly being higher than for phonetic algorithms but lower than for Edit distance ones. The Token algorithm is an interesting intermediate between the flexible N-gram algorithms and the precise string match matching? It can be particularly interesting when spelling errors are not frequent in the different data files to be matched. Tokens from *Patrick Aebischer* lead to a perfect match with the tokens from *Aebischer Patrick*. Unlike simple string matching, Soundex or Metaphone, the algorithms Edit distance, N-gram and Token provide a scale of similarity (or distance) between 0 and 1.

The different algorithms are usually customized to improve their performances. This explains why there are so many competing algorithms compared and crafted by researchers. Among possible improvements, the weighting procedure is appealing: it gives more importance to observations or changes that are less likely to occur in a text. Weights can thus easily be applied to Edit transformations or to N-grams and Token vector elements. For instance, in an Edit distance algorithm, a higher weight may be positioned for replacing the character r with the character d which is closer than for example the character m on a QWERTY keyboard. Different choices on weights lead to different parent algorithms (e.g. the Smith-Waterman distance, the Hamming distance). In N-gram or Token algorithms, grams or tokens more present in inventors names and addresses are less informational that rare grams or tokens. A simple approach is thus to weight grams or token assigning them a weight equal to the inverse number of their occurrences in the database.

Several rankings of matching algorithms are already available in the literature on name matching (See Pfeifer *et al.*, 1996; Zobel and Dart, 1995; Phua *et al.*, 2006). To our knowledge there is however no available comparisons have been obtained matching personal names and patent data. Furthermore, a clear hierarchy is hard to achieve here for several reasons: firstly, the relative performances depend on the type of strings to match. Differences do exist between results obtained on last and first names (Pfeifer *et al.*, 1996; Christen, 2006; Phua *et al.*, 2006), on female or male first names (Phua *et al.*, 2006). Secondly, performance is subjective since

several criteria are taken into account (recall, precision, computational cost). Due to the size of patent data, computation can indeed be a serious hampering factor for many algorithms (the Edit distance variants for example). Thirdly, as mentioned earlier, the matching stage depends on the parsing stage. Some matching algorithms cannot be efficient with a too strong data cleaning: for example, the deletion of spaces between words (as in Mageman *et al.*, 2006) creates large strings which cannot be handled by Soundex or Token algorithms. Some algorithms perform however better than others: Phonex or 2-gram are found to be better performers than 3-gram, 4-gram, or Damerau-Levenshtein algorithms (Pfeifer *et al.*, 1996; Phua *et al.*, 2006; Christen, 2006). According to the surveyed literature, hybrid matching algorithms have better results (*e.g.* Zobel and Dart, 1995; Pfeifer *et al.*, 1996; Hodge and Austin, 2003; Phua *et al.*, 2006). Examination of the various kinds of complex matching algorithms is beyond the scope of this paper even if we have explored the outcomes of some of them.

Although the matching stage is very alluring, it is nevertheless true that the parsing stage is not negligible, especially in multinational databases. Large multinational databases such as PATSTAT are filled by multiple national entities around the world, each having their own systems and protocols to do so. On the one hand, when assembling this data some information is likely to get corrupted. This is the particular case of special characters such as accentuated letters. On the other hand, some entities may have different protocols about the data entry process. For instance, some will consider all accentuated characters, some less and others none. Also, some entities will standardize the name as *Last Name, First Name* and others not. Also, a number of entities enter information manually while others use automatic optical recognition systems. A lot of progress has been made on the cleaning stage concerning firms' names in patent data (See especially Derwent, 2002; Hall, 2006; Mageman *et al.*, 2006) and even inventors' names (See Hoisl, 2006). Several text string parses are now well identified and can be applied to prepare the different data sets. Little is known however on the gains associated with the different parsing strategies and with each step applied. The same remarks hold for the filtering stage. The comparison between different filters is however seriously hampered by data scarcity and the subsequent restricted number of filters to compare.

### 3. Data sources and benchmark set

The datasets we are using to perform the tests on matching algorithms are from three different sources. The first and largest dataset is the September 2006 version of PATSTAT and contains approximately 12 millions inventors' names who filed a patent application at either the EPO or the United States Patent and Trademark Office (USPTO). The second source is the list of inventors registered at the patent portfolio held by the *Service des Relations Industrielles* from the EPFL. This list contains the 1995-2005 EPFL inventors defined as any inventor(s) or co-inventor(s) of a declared invention made at the EPFL. These 841 EPFL inventors are more likely to be registered in EPO or USPTO records over the period<sup>4</sup> since most EPFL inventions are patented. Finally, a third source of information is the 1994-2006 EPFL employee register provided by the EPFL Human Resource services. This register provided us with 8,885 non-

---

<sup>4</sup> Eventually, these inventors may have applied to patent offices other than EPO or USPTO without claiming priority in these latter.

administrative different EPFL names. Among the 841 EPFL inventors, 515 were employed for at least during one year over the period.

The EPFL list of inventors is of particular interest for several reasons: firstly, the EPFL list is a sample of names from various nationalities and cultures due to its geographic position and important immigration flows of scientists. Secondly, the TTO list provides additional information on inventors: name, surname, middle name (not systematic, NS thereafter), personal addresses (NS), Scientific lab (NS), co-inventors with its personal address (NS), co-ownership, licensees ID including their personal addresses. Thirdly, the information on inventors is complemented by the annual Human Resource list for employees' name, surname, middle name (NS), gender and the different ZIP codes of personal addresses over the period. Thanks to one's birth date or the EPFL individual code (the two authors of this article are respectively N°178326 and N°168647), the human resources list also offers a clear separation among homonymic researchers and potential changes in names. Changes in family names due to marriages or divorces should be scarce as the EPFL is an engineering school where female teachers or researchers are still a minority. Finally, the TTOs list is an interesting pre-selection of inventors. If all patents until 2005 are available, names should match by 100%. The only caveat here is that recently filed patents are not yet available in PATSTAT and that some inventions are not protected, especially on software.

Mixing information on inventors and on patents, implemented manually or applying the different filters and algorithms listed previously, checking manually a large list of matches for possible errors, we succeed in identifying 374 EPFL inventors having at least one patent application in PATSTAT, meaning a total of 2,607 pairs of names. After this first step, around 777 pairs remain ambiguous. For example, a researcher is found to have the same name and surname but with an address in a different country: the EPFL researcher indeed spent 6 months in Sweden as an invited professor. The information is however available from nowhere unless by directly contacting the researcher. We decided to keep our benchmark set of patents imperfect but to clean for these ambiguities. Not going further on these 777 pairs allows us to keep a clear distinction in our results between what is available without heuristics (*e.g.* perfect match) and what cannot be solved by such heuristics without additional information, something which is usually not available from existing datasets. In conclusion, our core benchmark set is composed of 1,830 pairs representing 349 EPFL researchers and their patents filed in EPO or USPTO.

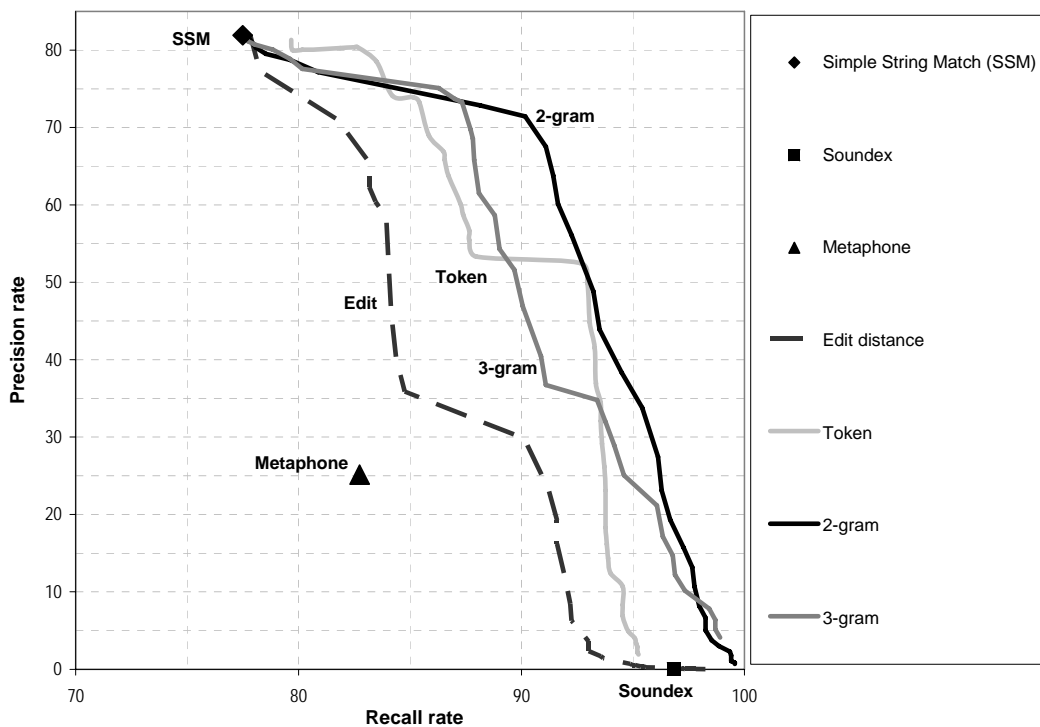
## 4. The matching stage

We first compare the following different algorithms: simple string match, Soundex, Metaphone, Edit distance, 2-Gram, 3-Gram and token algorithms applied to our multi-parsed benchmark set. The particular Soundex function we are using here is the one described by Knuth (1973: 391-392), and the particular Metaphone function is the one described by Binstock and Rex (1995). Finally, we transform the Levenshtein distance result dividing it by the maximum length of both

text strings and subtracting this result from the unity<sup>5</sup>. N-grams and Token algorithms are implemented in a weighted fashion<sup>6</sup> as the non-weighted ones are found to be dominated strictly by their weighted versions (results available upon request).

Our main results are represented in Figure 2, where the Recall-Precision results for simple string match, Soundex and Metaphone are represented by single points and for Edit distance, N-grams and Token which are represented by decreasing curves. These curves characterize the Recall-Precision trade-off when changing the value of the algorithm similarity threshold. The closer the threshold is set to one the higher is the Precision rate, while the lower it is set, the higher is the Recall rate.

**Figure 2: The impact of different algorithms on the precision-recall frontier, using already multi-parsed string**



The simple string match is supposed to be the best algorithm regarding the precision rate (82%), whereas the recall rate is naturally low at 77% minimizing false positive cases. From the precision rate point of view, this algorithm is however dominated by the Token algorithm

<sup>5</sup> The relative similarity reformulation is  $1 - \frac{D}{\text{Max}(l_1, l_2)}$  where D is the Levenshtein distance expressed in number of operations and  $l_i$  is the number of characters of the text string i.

<sup>6</sup> For each gram or Token, the weight is computed as:  $\frac{1}{(\log n_i + 1)}$  where n is the number of occurrences in PATSTAT of the token or gram i.

providing the same precision rate but with more than 79% as a recall rate. It also highlights that other algorithms such as N-gram weighted algorithms are the best performers and are equivalent to the simple string match when the maximum precision is targeted.

Conversely, it is shown that the usual Soundex algorithm<sup>7</sup> accepts almost every pair with a quasi-null precision rate. The use of Soundex practically fully postpones the identification of false positive inventors to a second matching stage – as in Trajtenberg *et al.* (2006) using simple string match on first names – or to the next filtering stage. As Figure 2 exhibits, the Soundex is strictly dominated by the N-gram algorithms: the same recall rate is reached with a better precision especially with the 2-gram algorithm.

The Edit algorithm is found to be equivalent to the Soundex even if Edit performs better in precision terms when we accept losing some recall rate. Surprisingly the Metaphone performs quite badly here and is dominated by Edit, N-gram and Token algorithms, although in terms of precision it seems to perform much better than Soundex.

The 3-gram is found at first sight to be slightly superior when precision is targeted. Below a 74% precision rate however the 2-gram becomes dominant. As already mentioned, the Token algorithm ability to manage name and surname permutations keeps the precision rate near to that of the simple string perfect matching algorithm and also provides better recall. However, the decay in precision starts quickly and the Token algorithm is dominated by weighted N-gram algorithms when the targeted recall rate is over 82%. In order to test the idea that mixed or hybrid algorithms are more efficient than single ones we combined the Token algorithm with both phonetic ones. The Soundex-Token and the Metaphone-Token algorithms – both weighted-were tested, resulting in improved performances of the phonetic algorithms. However, these mixed algorithms – not reported in Figure 2, but available upon request – are still completely dominated by the Token and N-gram algorithms.

Strictly in terms of the matching stage, we conclude that when precision is targeted by scholars, the weighted Token algorithm is a dominant choice. Instead, when the researchers aim at a general identification of patent portfolio rather than a sampling view and agree to give up some precision in order to reintegrate false negatives, the weighted 2-gram algorithm is a good choice. As shown in Figure 2, in the latter case, the potential decrease of the precision rate is around 10% whereas the recall rate increases up to 13%. The weighted 2-gram strategy is also interesting since it is more robust than other algorithms in errors associated with fixing the distance threshold.

As a result, in the following sections we will focus mainly on the 2-gram and Token algorithms.

---

<sup>7</sup> We use here the original Soundex, which retains only the first four characters. When retaining 8, 16 or 32 characters the recall rate drops steeply while the precision rate does not improve greatly. Results are available upon request.

## 5. The parsing stage

We propose in this section to explore the different data preparation strategies that can be implemented in a parsing stage. Some problems can be treated straightforwardly by quite simple parsing steps while others cannot. A middle name for example is additional information in patent data sets (21% in a PATSTAT subsample) which can create serious problems when algorithms are not flexible. The same remark applies for different spellings, inverted names and surnames or any additional noisy information added into the name field (e.g. address, institution, title). Conversely, some other problems such as different case (in 71% of cases), symbols (18%), accentuated characters (15%), double spacing (14%) are frequent and easy to deal with through systematic parsing when applied to the two datasets to match.

*Figure 3: The impact of different parsing tasks on the precision-recall frontier, for weighted 2-gram algorithm*

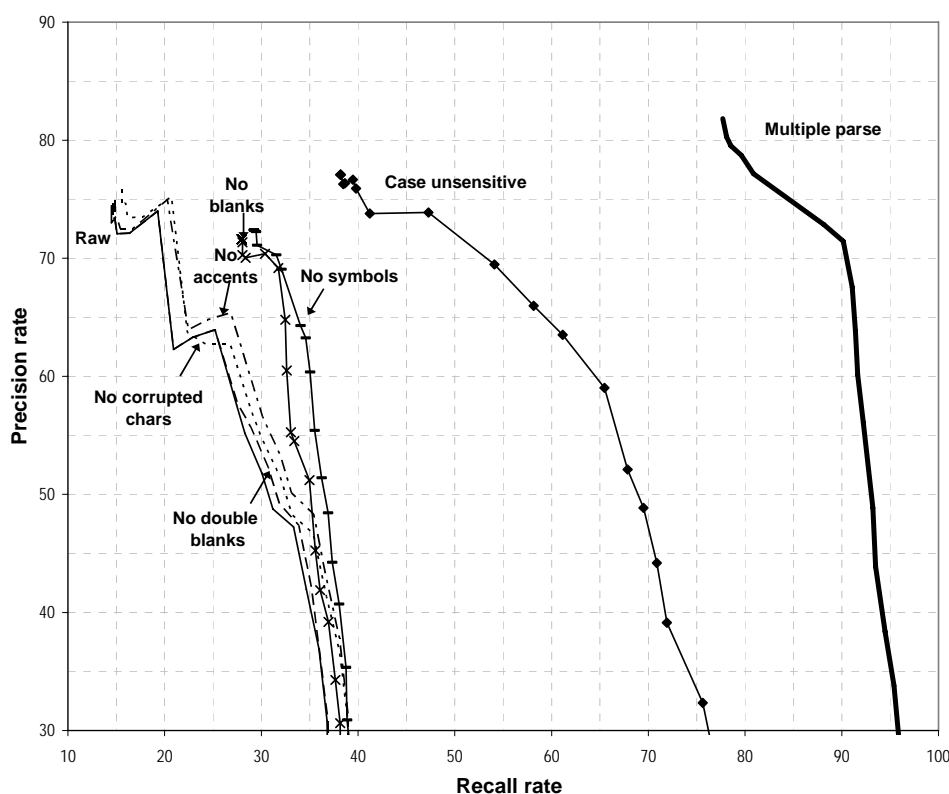


Figure 3 highlights that some parsing steps are much more interesting than others by comparing the five precision-recall curves to the original one (Raw curve in Figure 3). The first parsing to apply is the transformation to the same case (all lower or all upper case), suggesting that the strong heterogeneity among observations and countries dominates the interest to keep the information contained in different cases, such as initials. This parsing procedure prevails even if the cleaning of symbols also brings an important increase in recall rates with a minor decrease

in precision. More unexpected are the improvements obtained from removing spaces, as a weighted 2-gram should minimize the space problem. Finally, removing accents, double blanks or other corrupted characters are parsing tasks which – if applied separately – do not greatly improve precision or recall.

Evidence also suggests the existence of synergies in applying several parsing strategies altogether. As can be seen in Figure 3, the rightmost curve reflects the results of applying all parsing strategies (but No blanks) and its gains in terms of precision and recall are much greater than that provided by the simple addition of each parsing marginal gain. The sequence of parsing steps we propose here induces an improvement in both precision (+7%) and recall rates (+73%) suggesting that scholars working on patent matching are pertinent when they insist in their papers on the importance of the parsing stage.

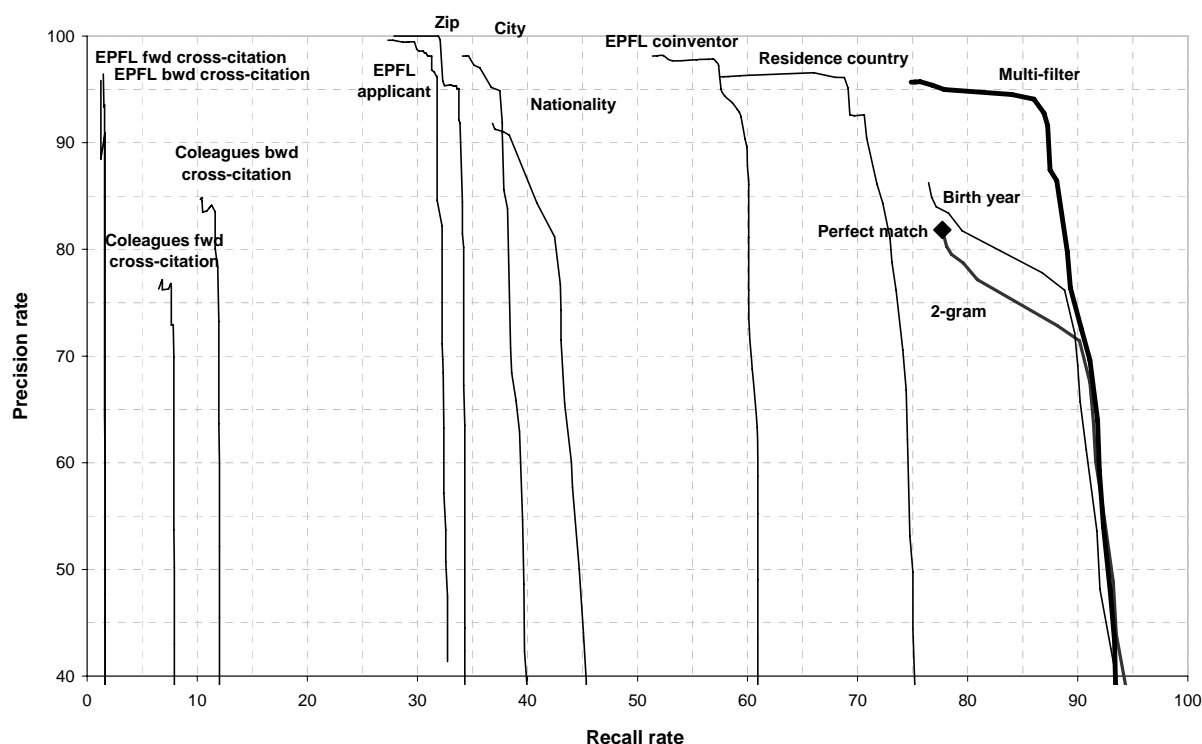
## 6. The filtering stage

The filtering stage depends on the ability to obtain and implement (through different algorithms or filters), complementary information on individuals in order to restore false negatives and separate false positives. Information such as Name and Surname, Affiliation or Field of research is often at hand for PhDs or inventors' lists (*e.g.* Mariani *et al.*, 2007). Hoisl (2006) complements her list of European inventors' names with last names, addresses, first IPC code or the ID of applicants. Additional filters are possible such as the exploitation of patent data on self-citations, co-inventors, similarity in citations (*e.g.* in Trajtenberg *et al.* 2006). Of course, the more information available, the higher the likelihood of improved precision and/or recall rates will be. However, the disambiguation procedures are complex: the *ex ante* sorting of criteria is not straightforward (See Hoisl, 2006) and the relative efficiency of each of these criteria and how to combine them are still open questions.

We thus confront the matches obtained within the matching stage with the different additional information contained in our EPFL benchmark set. This complementary information can be divided by its nature in two groups: the information regarding each inventor and the information regarding the characteristics of the total list (in our case the EPFL). Regarding complementary data on inventors, we have data on their country of residence, city and ZIP code, their nationality and their date of birth. While this last piece of information can be used only to exclude patents filed outside a most likely patenting life period (*e.g.* between 20 and 70 years old), the other complementary data can be used to check the address and country fields from each potential patent file matched.

Regarding the institution, there is additional information available on the institution itself such as all its different spellings (*e.g.* EPFL or École Polytechnique Fédérale de Lausanne) or the different addresses of the units or departments, which can also be checked against the address and country fields. But there is also the remaining list of EPFL employees where all individuals are possible co-inventors or inventors of other patents, thereby cross-citing the potential matched one.

**Figure 4: The impact of different disambiguation filters on the precision-recall frontier, for 2-gram weighted algorithm**



As in the previous sections, we first tried out each filter separately in order to estimate the marginal gain of each when applied to the potential matches. When a single filter is applied, the risk of cutting off correct positives also exists, which would at the same time decrease both recall and precision rates. This is especially the case when the filter is applied on fields where missing data are not very infrequent.

Figure 4 presents the marginal gains of the different single filters as well as a multiple one applied after a weighted 2-gram matching process<sup>8</sup>. Results confirm that filters applied individually may cut-off too many correct positive matches, as all our results (except for the Birth year filter) are shifted towards the left with respect to the original weighted 2-Gram. Nevertheless, results also show that most filters may offer at least some precision gain when compared to the original weighted 2-Gram. Interestingly, Nationality seems much less effective than Country of Residence, implying that in most of the cases patents are correctly filed using the inventors' address instead of their nationality. Furthermore, both forward and backward citations are much less likely to provide disambiguation than other filters, especially for EPFL co-inventors.

<sup>8</sup> Some weights could be introduced for filters. For example, Trajtenberg *et al.* (2006) give more weights to small cities since the likelihood of finding two homonymic inventors in the same city is lower.

The results from applying a multiple filter including all single filters (but Birth year, Nationality and Cross-citations) are exposed at the rightmost side of Figure 4. It is clear that there is room for improving both precision (+13%) and recall (+11%) after applying the correct disambiguation strategy.

## 7. Conclusion

Patent data are now extensively used by scholars. Furthermore, patent data are increasingly matched with other lists of firms' names or lists of inventors' names in order to rebuild patent portfolios. Little is known however about the reliability of the procedures implemented by researchers. Trajtenberg *et al.* (2006) unveiled the importance and the complexity of this names game and proposed a single heuristic. We still do not know what the best heuristics to apply are when using different parsing, matching and filtering stages. Using the PATSTAT dataset and an EPFL benchmark list of inventors, we explore several solutions. A first general result is that parsing, matching and filtering are three important stages and that, the interactions between stages do not allow any disequilibrium among them in order to achieve high Precision and Recall rates. More precisely, our findings can be summarized as follows:

The *parsing stage* is mostly important when raising the recall rate is required. As already identified by several scholars, a sequential multi-parse procedure is the most efficient way to proceed. Even if some parsing strategies (case insensitiveness, removing symbols) are more productive than others (e.g. removing accents), it is their interaction which provides the greatest improvement. According to our results, we consider that the best parsing strategy is to apply the following sequence of five different parsing steps:

- (1) Correct all corrupted characters by replacing them with the most likely character.
- (2) Replace all accentuated or non Latin characters by their non accentuated or Latin version, such as *é* to *e* or *ß* to *ss*.
- (3) Replace all symbols – including numbers and punctuation marks which are not in the 26 letter Latin alphabet - with blank spaces.
- (4) Eliminate all double spacing and all heading or trailing blank spaces.
- (5) A final step is the general conversion of every name field to the same case (either uppercase or lowercase).

Applying a more sophisticated algorithm in the *matching stage* enables improving the recall rate. A simple string match (SSM), which is usually applied by scholars, is outperformed by most of the explored algorithms (see Figure 5). Our results suggest that:

- (1) When precision is targeted by scholars, the weighted Token algorithm is a dominant choice, although closely followed by the weighted 2-gram.
- (2) When the researchers aim at a general identification of patent portfolio rather than a sampling view and agree to give up some precision in order to reintegrate false negatives, the weighted 2-gram algorithm seems to be the best choice. Of course a potential decrease of the precision rate is acknowledged in order to obtain a better recall rate.

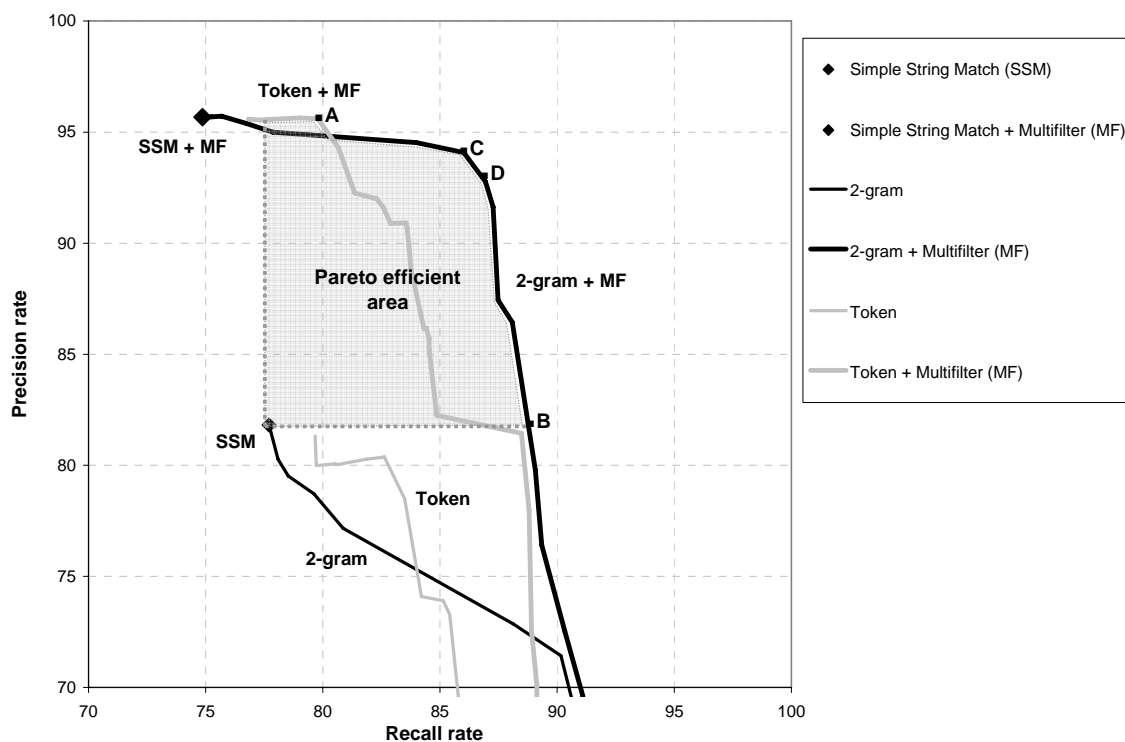
The *filtering stage*, sometimes neglected, is a good way of substantially improving the precision of the matching procedure. Our results suggest that scholars using their additional information

on inventors and institutions could improve their precision rate by applying an appropriate filtering strategy. In the case of datasets with incomplete information – *e.g.* the country codes or addresses of inventors in PATSTAT – a single filter can be too restrictive, thus the best filtering strategy is the inclusion of alternative disambiguation filters. The following is a list of six different filters proven to be the best approach according to our benchmark set:

Keep all matched patents when...

- (1) the inventor’s residence country is the same as the patent one,
- (2) OR the inventor’s ZIP code is within the patent address field,
- (3) OR the inventor’s city name is within the patent address field,
- (4) OR the potential patent co-inventors’ names match any name within the inventor’s organization (if any),
- (5) OR the potential patent applicant matches any plausible spelling of the inventor’s organization (if any),
- (6) AND the potential patent was filed during a reasonable inventor’s life span (between 20 and 70 years).

**Figure 5: Recall and precision rates of weighted 2-gram and Token after applying a multiple disambiguation filter**



However, the proposed heuristic requires additional information to be implemented. The filtering stage, despite the enhancement it brings when introduced, does not alleviate the

tradeoff between precision and recall as underlined by the grey area in Figure 5 showing the results using a multi-filtered weighted 2-gram or Token matching algorithms<sup>9</sup>. Hence several options are possible here.

Scholars interested in precision will be keen to apply multi-filter on the Token matching results, where the highest precision is achieved (96%) at point A with a recall gain of 5% with respect to a simple string match (*ceteris paribus* the parsing and filtering stages). On the contrary, scholars aiming to maximize their recall can apply a multi-filter after a 2-gram to minimize the impact on the precision rate. For instance, in point B an improvement of 10% in the recall rate can be achieved without any loss on the precision rate if compared with the simple string match (*ceteris paribus* the parsing stage).

Figure 5 also suggests that the false negatives are likely to be much lower when it is conceded to leave the precision rate drop slightly. The weighted 2-gram after the multi-filter is quite flat for any similarity threshold between 1 and 0.9. Points C and D in Figure 5 represent respectively the thresholds 0.91 and 0.90.

However, three additional aspects of our results remain which may persuade scholars to implement these kinds of heuristics: their implementation, their impact on academic studies and their robustness.

The first aspect deals with the application of the sequence we propose as a solution. How can we apply the recipe to a dataset when no benchmark set is available? Two solutions are available here: the first one is to apply directly the different steps listed above, tuning the threshold in the algorithms in order to either obtain maximum precision or recall rate as desired. The solution is rapid, but also ignores the warning that our results rely on a specific list of names where Western European names are dominant. A second solution consists in adding a test on a benchmark set in order to adapt the heuristic to the potentially idiosyncratic names' list. To construct such a benchmark set, the use of different algorithms is possible; the idea is that the recall rate must be wide enough to be able to retrieve manually individuals who are false negative matched in a simple string match. Box 1 summarizes the steps for this second strategy.

***Box 1: Steps to follow to build your own benchmark set:***

1. Pick a small but statistically significant subset from your list of inventors. If possible try to select the most likely to patent. For instance, we have chosen those researchers who have already a patent declared at the university's TTO.
2. Apply to this subset the 5 different cleaning procedures as stated above.
3. Compute the distance of your benchmark inventors from the inventors in the dataset you want to match with (e.g. EPO's PATSTAT) using the weighted 2-gram algorithm.
4. Keep a wide threshold (such as 0.5) in order to select potential false negative patents.

---

<sup>9</sup> The multi-filter applied includes only the first five filters ( (1) to (5) ) suggested as it is not usually easy to acquire detailed information such as the birth date. We have tested both with and without this last filter and results are rather similar, with the expected increase in precision when using filter (6).

5. Apply different disambiguation filters according to any complementary information available.
6. Manually check the results on the subset sample.
7. Test the precision and recall, confronting your results obtained by hand with the ones obtained mechanically using the three stages. Determine the threshold of your interest, that is to say what values will satisfy you from a precision-recall point of view.
8. Apply Steps 2 to 6 to the remaining inventors' list using the threshold chosen in step 7.

Results – reported in the Appendix – suggest that the use of the different algorithms leads to different results. While the influence of several variables is found to be robust (age, foreigners), others, such as career level variables or scientific field are found to be heterogeneous in both coefficient and significance. The heuristic which deals with both Precision and Recall is the only one which correctly identifies the positive effect of tenured foreign professors. Worse, heuristics focused on precision only suggest that tenured professors are more likely to patent which is not true according to our benchmark set. Note that the results obtained on maximized recall are found to be a valuable solution. The volatility of results underlines the importance of the “Names Game” stage in patent based research. The results cast doubts on the relevance of precision as a target in the names game and suggest a lack of reliability of econometric results subsequently obtained.

A final question regards the *robustness* of our heuristics. Two dimensions are concerned here. The first one deals with the heterogeneity of the names' list. Our results depend on the structure of our benchmark set (Bilenko *et al.*, 2003). Even if the EPFL is one of the most important international schools in Europe, this does not guarantee that the sequence is reliable for Swedish or Japanese names. These tests clearly implicate the future extension of this work. The second dimension is the robustness of our results for firms' names, that is to say the other names game usually played by scholars. In order to test the robustness of our conclusions we applied the same different heuristics to the French list of firms which declared patents in the second French Community Innovation Survey (CIS2). The conclusions were similar to those obtained for inventor's names (Raffo and Lhuillery, 2008)<sup>10</sup>.

---

<sup>10</sup> Results are available upon request.

## 8. References

- Agarwal R., Ganco M. and Ziedonis, R.H. (2007) Reputations for Toughness in Patent Enforcement: Implications for Knowledge Spillovers via Inventor Mobility, WP., December.
- Azoulay P., Ding W. and Stuart T. (2007) The determinants of faculty patenting behavior: Demographics or opportunities? *Journal of Economic Behavior & Organization*, 63(4) pp. 599-623.
- Bilenko M., Mooney R.J., Cohen W.W., Ravikumar P. and Fienberg S.E. (2003) Adaptive Name Matching in Information Integration, *IEEE Intelligent Systems*, 18(5) pp. 16-23.
- Binstock A. and Rex J. (1995) *Practical Algorithms for Programmers*, Addison-Wesley Longman Publishing Co., Inc., Boston, MA.
- Bound J., Cummins C., Griliches Z., Hall B.H. and Jaffe A.B. (1984) Who Does R&D and Who Patents? In *R&D, Patents and Productivity*, edited by Zvi Griliches. Chicago: University of Chicago Press, pp. 21-54.
- Cassiman B, Glenisson P. and Van Looy B. (2007) Measuring industry-science links through inventor-author relations: A profiling methodology, *Scientometrics*, 70(2) pp. 379 - 391.
- Christen P. (2006) A Comparison of Personal Name Matching: Techniques and Practical Issues In proceedings of the Workshop on Mining Complex Data (MCD), IEEE International Conference on Data Mining (ICDM), Hong Kong, December.
- Damerau F.J. (1964) A technique for computer detection and correction of spelling errors. *Communications of the ACM*, 7(3) pp. 171-176
- Derwent world patents index patentee codes, Revised Edition 8, 2002, Thomson Scientific, UK, 293p. (<http://www.thomsonscientific.com/media/scpdf/patenteeCodes.pdf>).
- Fleming L., King C. and Juda A. (2004) Small Worlds and Innovation, Working paper, Harvard Business School, August, 32p.
- Hall P.A.V. and Dowling G.R. (1980) Approximate String Matching, *ACM Computing Surveys*, 12(4) pp. 381-402.
- Hall B. (2006) The Patent Name-Matching Project, <http://elsa.berkeley.edu/~bhhall/pat/namematch.html>.
- Hodge V.J., Austin J. (2003) A comparison of standard spell checking algorithms and a novel binary neural approach, *IEEE Transactions on Knowledge and data engineering*, 15(5) pp. 1073-1081.
- Hoisl K. (2006) German PatVal Inventors – Report on Name and Address-Matching Procedure, March.
- Holmes D. and McCabe D.C. (2002) Improving Precision and Recall for Soundex Retrieval, 2002 International Symposium on Information Technology (ITCC 2002), 8-10 April, Las Vegas, NV, USA. IEEE Computer Society.
- Jaffe B. (1986) Technological opportunity and spillovers of R&D: evidence from firm's patents, profits, and market value. *American Economic Review* 76(5) pp. 984-1001.
- Kerr W.R. and Fu S. (2007) The Industry R&D Survey – Patent Database, Link Project, Harvard Business School WP-031, November.
- Kim J. and Marschke G. (2006) Proposal for the Use of NSF's SED and SDR Data Sets, Prepared for the "Using Human Resource Data from Science Resources Statistics" Workshop, September.
- Kim J., Lee S.J. and Marschke G. (2006) International Knowledge Flows: Evidence from and inventor-firm matched data set, Working paper, February, 37p.
- Knuth D. (1973), *The Art of Computer Programming*, vol. 3: Sorting and Searching, Addison-Wesley, pp. 391-392.
- Levenshtein V.I. (1966) Binary codes capable of correcting deletions, insertions, and reversals *Soviet Physics Doklady*, 10(8) pp. 707-710.
- Lissoni F. and Montobbio F. (2006) Inventorship attribution in academic patents: quantitative analysis of patent-publication pairs. WP, 2nd draft, June.
- Magerman T., Van Looy B. and Song X. (2006) Data production methods for harmonised patent statistics: patentee name harmonization, K.U. Leuven FETEW MSI Research Report 0605, Leuven, March, 88p.
- Mariani M. and Romanelli M. (2007) "Stacking" and "picking" inventions: The patenting behavior of European inventors. *Research Policy*, 36(8) pp. 1128-1142.
- Marx M., Strumsky D. and Fleming L. (2007) Noncompetes and Inventor Mobility: Specialists, Stars, and the Michigan Experiment, Harvard Business School W.P., February 1.
- Pfeifer U., Poersch T. and Fuhr N. (1996) Retrieval effectiveness of proper name search methods, *Information Processing & Management*, 32(6) pp. 667-679.
- Philips L., (2000) The Double Metaphone Search Algorithm, *C/C++ Users Journal*, June. (<http://www.ddj.com/cpp/184401251>).
- Phua C., Lee V. and Smith K. (2005) The Personal Name Problem And a Recommended Data Mining Solution, Working paper, School of Business Systems, Faculty of Information Technology, Monash University, Australia.

- Raffo J. and Lhuillery S. (2008) Matching procedures and harmonization methods, Workshop on Harmonisation of Applicants' names in Patent Data, OECD, 13<sup>th</sup> of march, Paris La Defense.
- Singh J. (2005) Collaborative Networks as Determinants of Knowledge Diffusion Patterns Management Science 51(5) pp. 756-770.
- Thursby, J., Fuller A. and Thursby M., (2007) US Faculty Patenting: Inside and Outside the University, NBER Working Paper Series N°13256, July.
- Trajtenberg M., Shiff G. and Melamed R. (2006) The "Names Game": Harnessing Inventors' Patent Data for Economic Research, NBER working paper series, No. w12479, September.
- Zobel J. and Dart P. (1995) finding approximate matches in large lexicons, Software - practice and experience, 25(3) pp. 331-345.

## Appendix

We introduce a negative binomial model for panel data with fixed effects in order to explain the number of patents filed each year per EPFL inventors. A negative binomial regression model is an extension of the Poisson regression model which allows the variance of the process to differ from the mean  $\lambda$  such that  $\lambda_i = \beta x_i + \varepsilon_i$  where  $\varepsilon_i$  is an introduced and unobserved individual factor where  $\exp(\varepsilon)$  often has a gamma distribution. The resulting probability distribution is:

$$\Pr(y_i / X_i) = \frac{\Gamma(\theta_i + y_i)}{y_i! \Gamma(\theta_i)} \left( \frac{\lambda}{\theta_i + \lambda_i} \right)^{y_i} \left( \frac{\theta_i}{\theta_i + \lambda_i} \right)^{\theta_i}$$

where  $\theta = 1/\alpha$  with  $V(y_i / X_i) = \lambda_i(1 + \delta\lambda_i)$ . The negative binomial model is modified to allow fixed effects in a panel. In this case, we have:  $\ln \lambda_{it} = \alpha_i + \beta' x_{it} + \varepsilon_{it}$  where  $\alpha_i$  is the individual fixed effect. The explanatory variables used in the empirical model (Table 1) are standard in the literature on academic patenting (see, for example, Azoulay *et al.*, 2007).

**Table 1: Negative Binomial regression on inventors' patents (USPTO and EPO), according to the matching strategy**

	(1) Benchmark ("real")	(2) Only Parsed (P+SSM)	(3) Parsed & Filtered (P+SSM+F)	(4) Highest Precision (P+Token+F)	(5) Balanced R&P (P+2-gram+F)	(6) Highest Recall (P+2-gram+F)
Age	0.292*** (0.066)	0.234*** (0.068)	0.200*** (0.062)	0.232*** (0.061)	0.225*** (0.061)	0.233*** (0.058)
Age squared	-0.003*** (0.001)	-0.003*** (0.001)	-0.002*** (0.001)	-0.003*** (0.001)	-0.003*** (0.001)	-0.003*** (0.001)
Female	-0.194 (0.534)	-0.308 (0.552)	0.196 (0.393)	0.277 (0.380)	0.112 (0.391)	-0.099 (0.385)
Single	-0.091 (0.254)	-0.004 (0.258)	-0.232 (0.217)	-0.233 (0.214)	-0.271 (0.212)	-0.133 (0.200)
Foreigner	-0.558** (0.253)	-0.654** (0.259)	-0.550*** (0.205)	-0.556*** (0.201)	-0.564*** (0.197)	-0.493*** (0.188)
Professor (tenured)	0.387 (0.327)	0.412 (0.347)	0.553* (0.297)	0.487* (0.291)	0.411 (0.286)	0.330 (0.282)
Professor (not tenured)	0.044 (0.228)	0.036 (0.245)	0.214 (0.212)	0.118 (0.208)	0.264 (0.199)	0.127 (0.194)
Foreign Professor (tenured)	0.704* (0.385)	0.340 (0.400)	0.181 (0.359)	0.437 (0.352)	0.576* (0.345)	0.547 (0.338)
Field: Architecture	0.355 (0.359)	0.408 (0.417)	0.881** (0.364)	1.121*** (0.348)	0.760** (0.332)	0.794** (0.324)
Field: Basic Sciences	0.627** (0.259)	0.803*** (0.270)	0.360* (0.202)	0.273 (0.197)	0.268 (0.195)	0.160 (0.188)
Field: Computer Sciences	0.128 (0.346)	0.205 (0.358)	0.061 (0.275)	0.027 (0.273)	0.007 (0.271)	-0.187 (0.267)
Field: Others	-0.368 (0.321)	-0.454 (0.384)	-0.459 (0.358)	-0.449 (0.342)	-0.207 (0.307)	-0.325 (0.304)
Constant	-6.650*** (1.445)	-5.632*** (1.482)	-6.235*** (1.353)	-6.833*** (1.332)	-6.583*** (1.315)	-6.515*** (1.254)
Fixed effects	Yes	Yes	Yes	Yes	Yes	Yes
Number of Inventor-years	2,050	1,887	1,856	1,886	1,962	1,985
Number of Inventors	267	246	242	246	254	258
Log-Likelihood	-1,195.00	-1,080.20	-1,339.95	-1,397.48	-1,457.04	-1,558.16
Chi squared	51.01***	41.22***	50.27***	65.37***	59.11***	54.48***

Notes: Standard errors reported in parenthesis; \*\*\* p<0.01, \*\* p<0.05, \* p<0.1;

P=Multiple Parse, F=Multiple Filter, SSM=Simple String Match, R&P=Recall and Precision.

Foreign Professor (not tenured) is not included since the variable is very collinear to the Professor (not tenured) variable.

Similarly, experience at the EPFL is also removed since it is very collinear to age.

Life science inventors are removed from the sample since the faculty only opened recently.

Engineering science is taken as a reference.

Note that the results in column (5) and (6) are obtained without any manual check. Such a step should make the results converge toward the results displayed in column (1).