Inventor Name Disambiguation for a Patent Database Using a Random Forest and DBSCAN

Kunho Kim[‡], Madian Khabsa[*], C. Lee Giles[†‡]

[‡]Computer Science and Engineering
[†]Information Sciences and Technology
The Pennsylvania State University
University Park, PA 16802, USA

[*]Microsoft Research
One Microsoft Way
Redmond, WA 98005, USA

kunho@cse.psu.edu, madian.khabsa@microsoft.com, giles@ist.psu.edu

ABSTRACT

Inventor name disambiguation is the task that distinguishes each unique inventor from all other inventor records in a patent database. This task is essential for processing person name queries in order to get information related to a specific inventor, e.g. a list of all that inventor's patents. Using earlier work on author name disambiguation, we apply it to inventor name disambiguation. A random forest classifier is trained to classify whether each pair of inventor records is the same person. The DBSCAN algorithm is use for inventor record clustering, and its distance function is derived using the random forest classifier. For scalability, blocking functions are used to reduce the complexity of record matching and enable parallelization since each block can be run simultaneously. Tested on the USPTO patent database, 12 million inventor records were disambiguated in 6.5 hours. Evaluation on the labeled datasets from USPTO PatentsView competition shows our algorithm outperforms all algorithms submitted to the competition.

Keywords

Name Disambiguation; Random Forest; DBSCAN

1. INTRODUCTION

Querying by person name is frequent in digital library search. For example, users may want to find all patents invented by certain person in a patent search system. If there is no unique identifier for each person in the database, processing this query can be problematic. In order to do this, there are several factors to consider. First, a person's name can be found in different formats from record to record. For example, in one record has the full name, "John Doe" and in another the first name initial and last name, "J. Doe". Second, many inventors share a common name and that name will appear often in the database. This problem is particularly significant for names from Asian countries. Statistics shown that 84.8% of the population have one of the top 100 popular surnames in China, while only 16.4% for the United States. Third, there exist typographical errors in names.

Name disambiguation algorithms are often used to solve these problems. Name disambiguation is the task of distinguishing each unique name from all name records in the database. Here, we propose to use an author, name disambiguation algorithm for the patent database. Our algorithm follows the typical steps of author name disambiguation, with newly proposed set of features from patent metadata. We train a random forest pairwise linkage classifier [5],[7], and use DBSCAN for clustering records [3]. The publicly available USPTO database is used for evaluation.

2. DISAMBIGUATION PROCESS

Patent records and scholarly publications have similar metadata, e.g. both with data on persons who published(or invented) and their affiliations. Ventura et al. [8] showed promising results in adopting author name disambiguation algorithms for the patent database. Our algorithm follows similar steps.

2.1 Random Forest Classifier

As in [7] we train a random forest(RF) classifier to determine whether each pair of inventor records is same person or not. The RF classifier is a well-known and popular ensemble learning method that combines simple decision trees by aggregating the votes from the trees for classification [1]. We started with the feature set used in Ventura et al. [8], and tested additional features that are used in author disambiguation. We only kept features that had a meaningful decrease in Gini importance if they were removed. Table 1 shows all features used. Our RF classifier consisted of 100 trees, with 5 features tried for each split. Testing was on two different datasets, Common characteristics and Mixture(see Section 3). The out-of-bag(OOB) errors of the RF were 0.05% and 0.07% respectively.

2.2 Blocking

Patent databases have a fairly large number of records; the USPTO database contains more than 12 million inventor record mentions. Blocking functions for preprocessing are crucial for scaling, especially for millions of entities. Blocking splits whole records into several blocks, and the clustering is done within each block separately, assuming records from the same person rarely split into different blocks. We

JCDL '16 June 19-23, 2016, Newark, NJ, USA

© 2016 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-4229-2/16/06.

DOI: http://dx.doi.org/10.1145/2910896.2925465

Table 1: Features used in the random forest

Category	Subcategory	Features
Inventor	First name	Exact, Jaro-Winkler, Soundex
	Middle name	Exact, Jaro-Winkler, Soundex
	Last name	Exact, Jaro-Winkler, Soundex, IDF
	Suffix	Exact
	Order	Order comparision
Affiliation	City	Exact, Jaro-Winkler, Soundex
	State	Exact
	Country	Exact
Co-author	Last name	# of name shared, IDF, Jaccard
Assignee	Last name	Exact, Jaro-Winkler, Soundex
Group	Group	Exact
	Subgroup	Exact
Title	Title	# of term shared

use a simple blocking function with *full last name+initial of first name*, so that we can easily parallelize the algorithm.

2.3 Clustering Using DBSCAN

DBSCAN, a density-based clustering algorithm[2], clusters inventor records. It does not require a prior the number of clusters, and it resolves the transitivity problem [3]. We use the fraction of negative(0) votes of the trees in random forest as the distance function.

2.4 Parallelization

The parallelization proposed in [4] using GNU Parallel [6] was used to utilize all available processing units. We assign each blocks to each thread. Memory limitations limit complete utilization of all CPUs. In our algorithm, the amount of memory required is proportional to total number of records in the block. As such, we divided all blocks into 3 groups based on the total number of records, and set different maximum threads to run simultaneously.

3. RESULTS ON THE USPTO DATABASE

Recently there was an inventor name disambiguation competition for the USPTO database. We used the same evaluation datasets to compare with the results of the competition. The training dataset includes the Mixture and Common characteristics datasets, and the test dataset includes ALS, ALS common, IS, E&S, Phase2. Detailed explanation for each dataset can be found on the competition's web page[1]. We measured pairwise precision, recall, and F1 score for evaluation.

Table 2 shows the results for each training and test dataset. Results were slightly better with the Common characteristics dataset, as expected from OOB error of RF. We can also see that the recall is relatively lower compare to the precision. Blocking effects the recall, as it can remove some potential matches. Table 3 shows the comparison between our work and the best result from the competition. Note our algorithm has the best performance on all datasets.

Table 2: Disambiguation evaluation

Test Set	Training Set	Precision	Recall	F1 Score
ALS	Mixture	0.9963	0.9790	0.9786
	Common	0.9960	0.9848	0.9904
ALS common	Mixture	0.9841	0.9796	0.9818
	Common	0.9820	0.9916	0.9868
IS	Mixture	0.9989	0.9813	0.9900
	Common	0.9989	0.9813	0.9900
E&S	Mixture	0.9992	0.9805	0.9898
	Common	0.9995	0.9810	0.9902
Phase2	Mixture	0.9912	0.9760	0.9836
	Common	0.9916	0.9759	0.9837

Table 3: Comparison with the competition winner

Test Set	F1(Ours)	F1(Winner)
ALS	0.9904	0.9879
ALS common	0.9868	0.9815
IS	0.9900	0.9783
E&S	0.9902	0.9835
Phase2	0.9837	0.9826

4. CONCLUSIONS

Our Random Forest DBSCAN author name disambiguation algorithm works very well for inventor names and readily scaled to over 12 million inventor name mentions. For efficient memory usage for scalability, better blocking functions would be useful. It would be interesting to see if other methods, such as graph or link data, could be incorporated as well.

5. ACKNOWLEDGMENTS

We gratefully acknowledge Evgeny Klochikhin and Ahmad Emad for helping us with evaluation and partial support from the National Science Foundation.

6. REFERENCES

[1] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[2] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining(KDD'96)*, volume 96, pages 226–231, 1996.

[3] J. Huang, S. Ertekin, and C. L. Giles. Efficient name disambiguation for large-scale databases. In *Proceedings of the 10th European Conference on Principle and Practice of Knowledge Discovery in Databases(PKDD'06)*, pages 536–544, 2006.

[4] M. Khabsa, P. Treeratpituk, and C. L. Giles. Large scale author name disambiguation in digital libraries. In *IEEE International Conference on Big Data*, pages 41–42, 2014.

[5] M. Khabsa, P. Treeratpituk, and C. L. Giles. Online person name disambiguation with constraints. In *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries(JCDL'15)*, pages 37–46, 2015.

[6] O. Tange et al. Gnu parallel-the command-line power tool. *The USENIX Magazine*, 36(1):42–47, 2011.

[7] P. Treeratpituk and C. L. Giles. Disambiguating authors in academic publications using random forests. In *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries(JCDL'09)*, pages 39–48, 2009.

[8] S. L. Ventura, R. Nugent, and E. R. Fuchs. Seeing the non-stars:(some) sources of bias in past disambiguation approaches and a new public tool leveraging labeled records. *Research Policy*, 2015.

[1] http://www.dev.patentsview.org/workshop