Vietnamese Author Name Disambiguation for Integrating Publications from Heterogeneous Sources

Tin Huynh, Kiem Hoang, Tien Do, and Duc Huynh

University of Information Technology, Vietnam,
Km 20, Hanoi Highway, Linh Trung Ward, Thu Duc District, HCMC
{tinhn,kiemhv,tiendv}@uit.edu.vn,
duc2802@gmail.com

Abstract. Automatic integration of bibliographical data from various sources is a really critical task in the field of digital libraries. One of the most important challenges for this process is the author name disambiguation. In this paper, we applied supervised learning approach and proposed a set of features that can be used to assist training classifiers in disambiguating Vietnamese author names. In order to evaluate efficiency of the proposed features set, we did experiments on five supervised learning methods: Random Forest, Support Vector Machine (SVM), k-Nearest Neighbors (kNN), C4.5 (Decision Tree), Bayes. The experiment dataset collected from three online digital libraries such as Microsoft Academic Search[1], ACM Digital Library[2], IEEE Digital Library[3]. Our experiments shown that kNN, Random Forest, C4.5 classifier outperform than the others. The average accuracy archived with kNN approximates 94.55%, random forest is 94.23%, C4.5 is 93.98%, SVM is 91.91% and Bayes is lowest with 81.56%. Summary, we archived the highest accuracy 98.39% for author name disambiguation problem with the proposed feature set in our experiments on the Vietnamese authors dataset.

Keywords: Digital Library, Data Integration, Bibliographical Data, Author Disambiguation, Machine Learning.

1 Introduction

In our previous work, we proposed and developed a system that is used to integrate the bibliographical data of publications in the computer science domain from various online sources into a unified database based on the focused crawling approach [9]. When we integrate information from various heterogeneous information sources, we must identify data records that refer to equivalent entities. One of the most important challenges for this problem is the author name disambiguation.

[1] http://academic.research.microsoft.com
[2] http://dl.acm.org/
[3] http://www.ieeexplore.ieee.org/

A. Selamat et al. (Eds.): ACIIDS 2013, Part I, LNAI 7802, pp. 226–235, 2013.

Table 1. The illustrative example, papers contained ambiguous author names

Paper no.1	*Multiagent Place-Based Virtual Communities for Pervasive Computing.* Conference: PERCOM '08 Proceedings of the 2008 Sixth Annual IEEE International Conference on Pervasive Computing and Communications Authors: ***Tuan Nguyen,*** Seng Loke; Torabi, T.; Hongen Lu. Dept. of Comput. Sci. & Comput. Eng., La Trobe Univ., Bundoora, VIC.	Mr. Tuan worked at Dept. of Comput. Sci. & Comput. Eng., La Trobe Univ., VIC., AU.
Paper no.2	*Stationary points of a kurtosis maximization algorithm for blind signal separation and antenna beamforming.* Journal: Journal IEEE Transactions on Signal Processing Authors: Zhi Ding, ***Tuan Nguyen*** Dept. of Electr. & Comput. Eng., Iowa Univ., Iowa City, IA.	Mr. Tuan worked at Dept. of Electr. & Comput. Eng., Iowa Univ., Iowa City, IA.
Paper no.3	*Semantic-PlaceBrowser: Understanding Place for Place-Scale Context-Aware Computing* Conference: The Eighth International Conference on Pervasive Computing (Pervasive 2010), Helinski, Finland, 2010. Authors: ***Anh-Tuan Nguyen***, Seng Wai Loke, Torab Torabi, Hongen Lu. Department of Computer Science and Computer Engineering, La Trobe University, Victoria, 3086, Australia	Mr. Tuan worked at Dept. of Comput. Sci. & Comput. Eng., La Trobe Univ., VIC., AU.

Name ambiguity is a problem that occurs when a set of publications contains ambiguous author names, i.e the same author may appear under distinct names (synonyms), or distinct authors may have similar names (polysems) [3]. The table 1 shows an example about Vietnamese author name ambiguity in 3 different publications. Author names in the paper 1 and the paper 3 are examples of synonyms. Both refer to 'Tuan Nguyen' from Department of Computer Science & Computer Engineering, La Trobe University, Melbourne, Victoria Australia. While author names in the paper 1 and the paper 2 are examples of polysems where 'Tuan Nguyen' in the paper 1 refers to 'Tuan Nguyen' from Department of Computer Science & Computer Engineering, La Trobe University, Melbourne, Victoria Australia and 'Tuan Nguyen' in the paper 2 refers to 'Tuan Nguyen' from the Department of Electric & Computer Engineering, Iowa University, Iowa City, IA, United States.

Name ambiguity in this context is a critical problem that have attracted a lot of attention of the digital library research community. Especially, authors with Asian names are very ambiguous cases that still be one of open challenges for this problem [3]. Therefore, we mainly focus on Vietnamese author name disambiguation for integrating the bibliographical data.

In section 2, we briefly present related research on developing digital libraries, building bibliographical database, disambiguating automatically author name. Section 3 presents our set of features that can be used to assist author name disambiguating in the context of bibliographical data integration of computer science publication from various online digital libraries. The experiments, evaluation and discussion will be introduced in section 4. We conclude the paper and suggest future works in section 5.

2 Related Work

Ferreira et al [3] did a brief survey of automatic methods for author name disambiguation. For this problem, the proposed methods usually attempt to group

citation records of a same author by finding some similarity among them (author grouping methods) or try to directly assign them to their respective authors (author assignment methods). Both approaches may either exploit supervised or unsupervised techniques. They also reviewed classification, clustering algorithms and as well as similarity functions which can be applied for this problem.

Han et al. applied a model based k-means clustering algorithm and a K-way spectral clustering algorithm to solve this problem [6][7]. Beside that, Han et al. also explored two supervised learning algorithms, Naive Bayes and Support Vector Machine. Both these supervised algorithms uses three types of citation attributes coauthor names, paper title keywords and journal title keywords, and achieved more than 90.0% accuracies in disambiguation [5]. In another research, Huang et al. presented an efficient integrative framework for solving the name disambiguation problem: a blocking method retrieves candidate classes of authors with similar names and a clustering method, DBSCAN, clusters papers by author [8]. For evaluation, they manually annotated 3,355 papers yielding 490 authors and achieved 90.6% for average accuracy.

Treeratpituk and Giles [11] also proposed random forest algorithm and a feature set for disambiguating author names in Medline dataset, a bibliographic database of life sciences and biomedical information. They applied and compared the random forests with some other algorithms such as NaiveBayes, Support Vector Machine, Decision Tree in this problem. Their experiments showed that the random forest outperforms than the others with the proposed feature set. The highest accuracy archived with random forest is 95.99% for the Medline dataset and their proposed feature set.

Qian et al proposed a labeling oriented author disambiguation approach, called LOAD, to combine machine learning and human judgment together in author disambiguation [10]. Their system have a feature that allows users to edit and correct author's publication list. All such user edited data are collected as UE (User Edited) dataset. They did experiments on their the UE dataset and DBLP dataset. The average accuracy archived 91.85% with their proposed method.

In general, author assignment methods mainly focused on learning a model for each specific author. They are usually very effective when faced with a large number of examples of citations for each author. However, DLs are very dynamic systems, thus manual labeling of large volumes of examples is unfeasible. In addition, authors often change their interesting area over time, new examples need be insert into training data continuously and the methods need to be retrained periodically in order to maintain their effectiveness [3].

While author grouping methods mainly focused on defining a similarity function used to group corresponding publications using clustering technique. The similarity function may be predefined or learned by using supervised learning methods. Learning a specific similarity function usually produces better results [3]. Therefore, in this research we proposed a set of features for learning a similarity function by using supervised learning methods.

3 Our Approach

How we can disambiguate these ambiguous names in two different publications. In order to do that, we can base on similarity of metadata of these publications. For example, similarity of names, affiliations, list of coauthors, keywords in publications which contain these ambiguous authors. Based on these metadata, we applied supervised learning methods and proposed a set of features that can be used to support training classifiers or learning a similarity function.

In this section, we present the popular string matching methods and consider to apply them for computing the similarity of metadata. We also present the proposed feature set for learning a similarity function.

3.1 Popular String Matching Measures

In [1][2], authors reviewed widely used measures in measuring similarity of two strings to identify the duplication. These measures basically are divided into three categories: (1) *Edit distance*; (2) *Token-based* and hybrid methods.

Edit Distance: distance between strings X and Y is the cost of the best sequence of *edit operations* that converts X to Y. There are some edit distance measures such as Levenshtein, Monger-Elkan, Jaro, Jaro-Winkler [2]. Levenshtein is one of the most popular measure for edit distance. For Levenshtein, the cost of converting X to Y is computed by three types of edit operations: (1) inserting a character into the string; (2) deleting a character into the string and (3) replacing one character with a different character.

$$Sim_{levenshtein}(X,Y) = 1 - \frac{d(X,Y)}{[max(length(X), length(Y))]}$$

Where:

- $d(X,Y)$ the minimum number of edit operation of single characters needed to transform the string X into Y.
- $length(x)$ the length of string X.

Token-Based Measures: in many situations, word order is not really important. In such cases, we can convert the strings X and Y to token multisets and consider similarity metrics on these multisets. Jaccard, TF/IDF [2], popular token based measures, widely used.

Hybrid Measures: Mogne-Elkan measure propose the following recursive matching scheme for comparing two long strings X and Y [2]. First, X and Y are broken into substrings $X = x_1...x_K$ and $Y = y_1...y_L$. Then, similarity is defined as:

$$Sim_{monge-elkan}(X,Y) = \frac{1}{K}\sum_{i=1}^{K}\max_{j=1}^{L} Sim'(x_i, y_j)$$

Where:

- Sim': is some secondary Edit Distance as Levenshtein, Jaro-Winkler.

In our problem, similarity of metadata strings (author name, co-authors, affiliation, keywords in publications) mainly based on tokens. Especially, most of ambiguous Vietnamese author names relate to the order of their family name and given name. The first order in full name sometimes is family name, while other cases are given name. Therefore, we have applied Jaccard method to calculate the similarity for features in our proposed feature set.

3.2 The Proposed Feature Set

To learn a similarity function, the disambiguation methods receive a set of pais of publications as the training dataset. The similarity of two ambiguous authors in a pair of publications is presented by a multidimensional vector. Therefore, we are going to pair publications that relate to ambiguous authors in our dataset. After that we encode similarity of this pair by a vector. In our dataset, for each pair of publications we labeled true (1) if two ambiguous authors are the same person or false (0) if two ambiguous authors are two different persons. We applied supervised learning algorithms to train, test and evaluate the performance of the proposed feature set. The detail of proposed feature set is described as following:

Author Name Similarity: We can assume that if the similarity of two string used to presented names of two ambiguous authors is high then two these ambiguous authors may relate to the same person. In order to calculate the similarity of names of these ambiguous authors, we used Jaccard coefficient.

$$Author\_Name\_Sim(A,B) = \frac{|Author\_Name\_A \bigcap Author\_Name\_B|}{|Author\_Name\_A \bigcup Author\_Name\_B|}$$

Where:

- $Author\_Name\_A$: name of author A presented in one specified publication.
- $|Author\_Name\_A \bigcap Author\_Name_B|$: number of same tokens in names of A and B.
- $|Author\_Name\_A \bigcup Author\_Name\_B|$: number of tokens in Author_Name_A or Author_Name_B.

For example:
Author_Name_A="Tuan Nguyen" and Author_Name_B="Nguyen Anh Tuan"
$|Author\_Name\_A \bigcap Author\_Name\_B| = 2$
$|Author\_Name\_A \bigcup Author\_Name\_B| = 3$
and Author_Name_Sim(A, B) = 0,6.

Affiliation Similarity: Affiliations of two ambiguous authors in two different publications is one of feature that can used to recognize whether two ambiguous authors actually mention to one person. In order to calculate the similarity of names of these affiliations, we also used Jaccard coefficient.

$$Aff\_Sim(A,B) = \frac{|Aff\_A \bigcap Aff\_B|}{|Aff\_A \bigcup Aff\_B|}$$

Where:

- Aff_A: Affiliation name of author A in one specified publication
- $|Aff\_A \bigcap Aff\_B|$: number of same tokens between affiliation of author A and affiliation of author B.
- $|Aff\_A \bigcup Aff\_B|$: number of tokens in string presented affiliations name of author A or author B.

CoAuthors Similarity: If two ambiguous author names share at least one same coauthor in two different publications then these ambiguous names may be one author. We proposed calculating method for the Coauthors_Name_Sim feature as following:

$$CoAuthors\_Names\_Sim(A, B) = MAX(Author\_Name\_Sim(A_i, B_j))$$

Where:

- $A_i \in CoAuthors(A)$
  and CoAuthors(A): is a set of co-authors of author A in publication P_1.
- $B_j \in CoAuthors(B)$
  and CoAuthors(B): is a set of co-authors of author B in publication P_2.

CoAuthor_Affs Similarity: If two ambiguous author names have coauthors who have worked in the same university or institute then these ambiguous names may be one author. We proposed calculating method for feature based on affiliation of coauthors as following:

$$CoAuthor\_Affs\_Sim(A, B) = MAX(Aff\_Sim(Aff_i\_P1, Aff_j\_P_2))$$

Where:

- $Aff_i\_P1 \in CoAuthors\_Affs(A)$
  i=1..n (n: number of coauthor of author A in P_1)
  and CoAuthors_Affs(A): is a set of affiliations of co-authors of author A in publication P_1.
- $Aff_i\_P2 \in CoAuthors\_Affs(B)$
  j=1..m (m: number of coauthor of author B in P_2)
  and CoAuthors_Affs(B): is a set of affiliations of co-authors of author B in publication P_1.

Paper_Keywords Similarity: If publications related to ambiguous authors contain similar keywords then two ambiguous authors may be the same person. We proposed the calculating method for the keyword-based feature as following:

$$Paper\_Keywords\_Sim(A, B) = \frac{|Paper\_Keywords\_A \bigcap Paper\_Keywords\_B|}{|Paper\_Keywords\_A \bigcup Paper\_Keywords\_B|}$$

Where:

- $Paper\_Keywords\_A$: is the set of keywords in publication A (publication of author A).
- $|Paper\_Keywords\_A \bigcap Paper\_Keywords_B|$: number of same tokens which keywords in publication A and publication B share.
- $|Paper\_Keywords\_A \bigcup Paper\_Keywords\_B|$: number of tokens of keywords in publication A or B.

4 Experiments and Evaluation

In order to analyze how classifiers and the feature set performed for the author name disambiguation problem, we applied the proposed feature set to many different classifiers such as Random Forest, kNN, SVM, C4.5, Bayes Nets. We did experiments to consider which classifier is to bring out higher accuracy and how classifiers and the proposed feature set effect to the author name disambiguation. This section presents our experimental results and discussions on archived results.

4.1 Dataset

Experimental dataset collected from three online digital libraries that are ACM DL, IEEE Xplore, MAS. In order to check the author name ambiguity when integrating publications from many various sources, we prepared 10 author names as the input data that used to submit to these digital libraries. All of these author names are Vietnamese author names (very ambiguous cases). For these authors, there are many different instance names. For example, author 'Kiem Hoang' can have many different names in the database such as 'Hoang Kiem', 'Kiem Hoang', 'Hoang Van Kiem', 'Kiem Van Hoang'.

Table 2. The dataset collected and labeled for training and testing

Submitted names	Number of publications collected	Number of pairs labeled	Number of pairs labeled with value 0	Number of pairs labeled with value 1
Cao Hoang Tru	30	435	26	409
Dinh Dien	30	435	51	384
Duong Anh Duc	30	435	29	406
Ha Quang Thuy	30	435	84	351
Ho Tu Bao	30	435	0	435
Kiem Hoang	30	435	25	410
Le Dinh Duy	30	435	57	378
Le Hoai Bac	30	435	0	435
Nguyen Ngoc Thanh	30	435	141	294
Phan Thi Tuoi	30	435	0	435
Total	**300**	**4350**	**413≅9.49%**	**3937≅90.51%**

For each author in this set, we collected 30 publications returned from 3 these online libraries. We do not care about the duplicate of publication from these libraries. These publications contain ambiguous author names for the integration (two different authors with the same name or one author with different names). We built the training and testing dataset by paring of publications in 30 publications for each author. Based on our understanding about these authors, we labeled for each pair with value 1 if ambiguous names in this pair actually be one

person and value 0 if these ambiguous names actually be two different persons. So, there are totally 4350 samples in our dataset. The table 2 show the detail of the dataset. Each sample in the dataset relate to a pair of publications which contain ambiguous authors. Each sample will be presented by a vector its each dimension is one specified feature.

4.2 Experiments for author disambiguation

We applied the k-fold cross validation, a method checks how well a model generalizes to new data, to evaluate the proposed feature set and various classifiers in our experiments. We used different supervised learning algorithms implemented by the WEKA project [4]. At this time, we tested with 5 various classifiers in WEKA such as kNN, Support Vector Machine, Random Forest, C4.5, Bayes Network. There are totally 4350 samples in the dataset.

The dataset is divided into 10 subsets with the same size. The cross-validation process is then repeated 10 times (folds). Each time, one of 10 subsets sequentially is used as test set and the remaining subsets are put together to form a training set. Therefore, each subset is used exactly once as the validation data. The 10 results from 10 folds then can be averaged to produce a estimation or an evaluation for classifiers. Experimental results, showing the accuracy of author name disambiguation with the proposed feature set and classifiers, are reported in table 3.

Table 3. The experimental results and evaluations with k-fold validation

k-Fold cross-validation	kNN	Random Forest	C4.5	SVM	Bayes
Validate fold-1	90.57	90.57	90.34	88.05	75.63
Validate fold-2	89.20	89.20	88.51	87.59	80.00
Validate fold-3	94.02	93.79	91.72	88.51	82.99
Validate fold-4	95.86	93.56	95.86	92.87	79.08
Validate fold-5	**98.16**	**98.16**	**98.39**	**97.93**	74.71
Validate fold-6	97.24	96.78	97.24	95.63	86.21
Validate fold-7	95.17	95.63	96.32	93.79	86.90
Validate fold-8	95.40	95.17	95.63	91.72	85.29
Validate fold-9	94.25	94.25	94.71	94.94	**88.05**
Validate fold-10	95.63	95.17	91.03	88.05	76.78
Average Accuracy	**94.55**	**94.23**	**93.98**	**91.91**	**81.56**

4.3 Discussion

Figure 1 compares the accuracy using of 5 classifiers based on the proposed feature set for the author name disambiguation problem. The figure presented

experimental results for all 10-folds cross-validation. The average accuracy of these 10-folds is shown in table 3. Our experiments shown that kNN, Random Forest, C4.5 classifier outperform than the others in most of folds (figure 1). The average accuracy archived with kNN approximates 94.55%, random forest is 94.23% and C4.5 is 93.98%. While the accuracy of SVM algorithm is 91.91%. Bayes classifier give lowest accuracy with this problem. The average accuracy archived with Bayes classifier approximates 81.56%. Summary, we archived the highest accuracy 98.39% for the author name disambiguation problem with our proposed feature set (table 3).

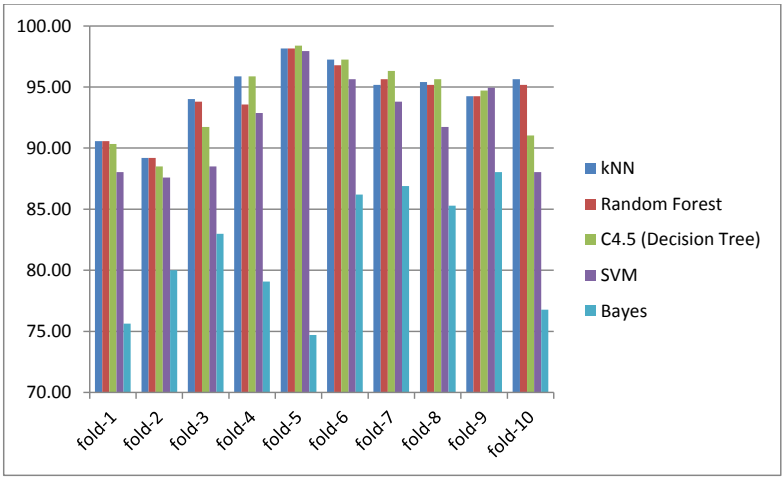

Fig. 1. The various accuracies produced by 10-folds cross-validation for author name disambiguation using different classifiers

5 Conclusion and Future Work

The gold of this research is to evaluate the performance of the proposed feature set which can be used to support training classifiers for disambiguating author names. Author name disambiguation is really a critical task for integrating computer science publications from various sources. We proposed and presented the feature set based on metadata that can be used to disambiguate author names in different publications. The feature set tested with 5 various classification algorithms. We built the dataset for training and testing based on publications contain Vietnamese authors. Because most of Vietnamese authors who often apply many different way to write names in their publications. The results show that our proposed feature set archived the average accuracy is 94.55% with kNN and the highest accuracy is 98.39% with the C4.5 algorithm (table 3).

In the future, we are going to continue improving the accuracy for this problem and doing more experiments for asian author name disambiguation and other

very ambiguous cases on DBLP, a public dataset. We will combine this module of author names disambiguation with our existing system that used to integrate computer science publications from various online sources.

Acknowledgments. This research is partially supported by VNU-HCM R&D project: B2011-26-01.

References

1. Bilenko, M., Mooney, R., Cohen, W., Ravikumar, P., Fienberg, S.: Adaptive name matching in information integration. IEEE Intelligent Systems 18(5), 16–23 (2003)
2. Cohen, W.W., Ravikumar, P.D., Fienberg, S.E.: A comparison of string distance metrics for name-matching tasks. In: IIWeb, pp. 73–78 (2003)
3. Ferreira, A.A., Gonçalves, M.A., Laender, A.H.: A brief survey of automatic methods for author name disambiguation. SIGMOD Rec. 41(2), 15–26 (2012)
4. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The weka data mining software: an update. SIGKDD Explor. Newsl. 11(1), 10–18 (2009)
5. Han, H., Giles, L., Zha, H., Li, C., Tsioutsiouliklis, K.: Two supervised learning approaches for name disambiguation in author citations. In: Proceedings of the 4th ACM/IEEE-CS Joint Conference on Digital Libraries, JCDL 2004, pp. 296–305. ACM, New York (2004)
6. Han, H., Zha, H., Giles, C.L.: A model-based k-means algorithm for name disambiguation. In: Proceedings of Semantic Web Technologies for Searching and Retrieving Scientific Data, Florida, USA (October 20, 2003)
7. Han, H., Zha, H., Giles, C.L.: Name disambiguation in author citations using a k-way spectral clustering method. In: Proceedings of the 5th ACM/IEEE-CS Joint Conference on Digital Libraries, JCDL 2005, pp. 334–343. ACM, New York (2005)
8. Huang, J., Ertekin, S., Giles, C.L.: Efficient Name Disambiguation for Large-Scale Databases. In: Fürnkranz, J., Scheffer, T., Spiliopoulou, M. (eds.) PKDD 2006. LNCS (LNAI), vol. 4213, pp. 536–544. Springer, Heidelberg (2006)
9. Huynh, T., Luong, H., Hoang, K.: Integrating Bibliographical Data of Computer Science Publications from Online Digital Libraries. In: Pan, J.-S., Chen, S.-M., Nguyen, N.T. (eds.) ACIIDS 2012, Part III. LNCS, vol. 7198, pp. 226–235. Springer, Heidelberg (2012)
10. Qian, Y., Hu, Y., Cui, J., Zheng, Q., Nie, Z.: Combining machine learning and human judgment in author disambiguation. In: Proceedings of the 20th ACM International Conference on Information and Knowledge Management, CIKM 2011, pp. 1241–1246. ACM, New York (2011)
11. Treeratpituk, P., Giles, C.L.: Disambiguating authors in academic publications using random forests. In: Proceedings of the 9th ACM/IEEE-CS Joint Conference on Digital Libraries, JCDL 2009, pp. 39–48. ACM, New York (2009)