A Hybrid Disambiguation Measure for Inaccurate Cultural Heritage Data

Julia Efremova[1], **Bijan Ranjbar-Sahraei**[2], **Toon Calders**[1,3]
[1]Eindhoven University of Technology, The Netherlands
[2]Maastricht University, The Netherlands
[3]Université Libre de Bruxelles, Belgium
i.efremova@tue.nl, b.ranjbarsahraei@maastrichtuniversity.nl,
toon.calders@ulb.ac.be

Abstract

Cultural heritage data is always associated with inaccurate information and different types of ambiguities. For instance, names of persons, occupations or places mentioned in historical documents are not standardized and contain numerous variations. This article examines in detail various existing similarity functions and proposes a hybrid technique for the following task: among the list of possible names, occupations and places extracted from historical documents, identify those that are variations of the same person name, occupation and place respectively. The performance of our method is evaluated on three manually constructed datasets and one public dataset in terms of precision, recall and F-measure. The results demonstrate that the hybrid technique outperforms current methods and allows to significantly improve the quality of cultural heritage data.

1 Introduction

Inaccurate information and lack of common identifiers are problems encountered when combining information from heterogeneous sources. There are a number of reasons that can cause inaccurate information such as spelling variations, abbreviations, translation from one language into another and modifying long names into shorter ones. Inaccurate information often occurs in many domains, for example, during information extraction from the Web or when attributing a publication to its proper author. Inaccurate information is very typical in cultural heritage data as well. In historical documents a real person could be mentioned many times, for instance in civil certificates such as birth, marriage and death certificates or in property transfer records and tax declarations. The name of the same person, his occupation and the place in such documents varies a lot. When working with such information, researchers have to identify which person references mentioned in different historical documents belong to the same person entity. This problem has been referred to in literature in many different ways but is best known as entity resolution (ER), record linkage or duplicate detection (Lisbach and Meyer, 2013; Christen, 2012; Bhattacharya and Getoor, 2007). The process of ER in historical documents is always accompanied by inaccurate information as well. As an example, there are more than 100 variants of the first name *Jan*, such as *Johan*, *Johannes*, *Janis*, *Jean* or the profession *musician* in historical documents can be spelled as *musikant*, *muzikant* or even *muzikant bij de tiende afd*. The latter means the musician in the 10th department.

The past few decades have seen a large research interest in the problem of inaccurate information. As a result, a large number of methods for comparing string has been developed. These standard methods are called string similarity functions. Some of those well known techniques are character-based, token-based or based on phonetic functions, for instance *Levenshtein Edit distance*, *Jaro Winkler distance*, *Monge Elkan distance*, *Smith Waterman distance*, *Soundex* and *Double Metaphone*. (Elmagarmid et al., 2007; Navarro, 2001; Winkler, 1995). Each of the mentioned similarity functions perform optimally for a particular dataset domain. For example, the phonetic function Soundex works great for encoding names by sound as it pronounced in English, but nevertheless sometimes it is also used to encode names in other European languages. However, only little work has been done in studying combinations of similarity functions, and in their simultaneous use for achieving more reliable results. Bilenko (2003) in his work computes names similarity with

Proceedings of the 8th Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities (LaTeCH) @ EACL 2014, pages 47–55,
Gothenburg, Sweden, April 26 2014.

affine gaps to train the Support Vector Machines classifier. Ristard and Yianilos (1998) designed a learnable Levenshtein distance for solving the string similarity problem. Tejada et al. (2002) learned weights of different types of string transformations.

In this paper we explore various traditional string similarity functions for solving data ambiguities and also design a supervised hybrid technique. We carry out our experiments on three manually constructed datasets: Dutch names, occupations and places, and also on one publicly available dataset of restaurants. The clarified function, that will allow us to recognize difficult ambiguities in textual fields, later will be incorporated into the overall ER process for a large historical database. The main contributions of this paper is a practical study of existing techniques and the design and the extensive analysis of a hybrid technique that allow us to achieve a significant improvement in results.

The remainder of this paper is structured as follows. In Section 2 we begin by presenting typical ambiguities in real-life cultural heritage data. in Section 3 we give an overview of standard string similarity functions. We describe the general hybrid approach in Section 4. In Section 5 we describe the prediction models that we use in the hybrid approach. In Section 6 we provide details about carrying out the experiments. In Section 7 we present an evaluation of the results. Section 8 offers a discussion about applying the designed approach to real-world data. Concluding remarks are given in Section 9.

2 A Real-Life Cultural Heritage Data

In this paper we use historical documents such as birth, marriage and death certificated provided by Brabants Historisch Informatie Centrum (BHIC) [1] to extract most common person names, occupations and places. Civil certificates are belonging to North Brabant, a province of the Netherlands, in the period 1700 - 1920. To study the name ambiguity we used a subset of data consisting of 10000 randomly selected different documents.Then for each name we obtain its standardized code in the database of Meertens Instituut[2] which has a large collection of Dutch names and last names and their typical variations. In the same way, in the database of The Historical Sample of the Netherlands (HSN) [3], for each occupation and place extracted from civil certificates where possible, we obtain its standardized code (van Leeuwen et al., 2002; Mandemakers et al., 2013). Historians have spent a number of years for creating a database of names, occupations and places variations. Using such data gives us a unique opportunity to explore typical variations in different domains and to design a robust technique which is able to deal with them automatically.

The resulting *Name* variations dataset contains 2170 distinct names that correspond to 1326 standardized forms. Table 1 shows a typical example of the constructed dataset of name variations.

ref_id	Name	name_id
1	Eustachius	1
2	Statius	1
3	Stefan	2
4	Stephan	2
5	Stephanus	2

Table 1: An example of a name variation dataset

The second dataset of *Occupations* contains 1401 occupation records which belong to 1098 standardized occupations.

The third dataset of *Places* contains 1196 locations records belonging to 617 standardized places.

3 Traditional Similarity Functions

There are three main different types of string similarity functions that can be used for variation tasks, namely character-based, phonetic-based and token-based. Each of them we investigate in detail below.

3.1 Character-Based Similarity

Character-based similarities operate on character sequences and their composition which makes them suitable for identifying imprecise names and spelling errors. They compute the similarity between two strings as the difference between their common characters. In this paper, we will consider the *Levenshtein edit distance* (LE), *Jaro* (J), *Jaro Winkler* (JW), *Smith Waterman* (SW), *Smith Waterman with Gotohs backtracing* (GH), *Needleman Wunch* (NW) and *Monge Elkan* (ME) string similarities (Elmagarmid et al., 2007; Christen, 2012; Naumann and Herschel, 2010). All of them return a number between 0 and 1 inclusively,

[1] http://www.bhic.nl

[2] http://www.meertens.knaw.nl/nvb/

[3] http://www.iisg.nl/hsn/data/occupations.html

where the highest value when two names are exactly the same. Table 3 shows an example of computed character-based similarities for three name pair-variants.

3.2 Phonetic-Based Similarity

Phonetic similarity functions analyze the sounds of the names being compared instead of their spelling differences. For example, the two names *Ste**f**an* and *Ste**ph**an* barely differ phonetically, but nevertheless they have different spellings. Phonetic functions encode every name with phonetic keys based on a set of rules. For instance, some algorithms ignore all vowels and compare only the groups of consonants, other algorithms analyze consonant combinations and thier sound that describe a large number of sounds. In this paper, we analyze 4 phonetic functions: *Soundex* (SN), *Double Metaphone* (DM), *IBMAlphaCode* (IA) and *New York State Identification and Intelligence System* (NY) (Christen, 2006). The Table 2 shows an example of applied phonetic keys to encode imprecise names.

Name	SN	DM	IA	NYSIIS
Stefan	S315	STFN	00182	STAFAN
Stephan	S315	STFN	00182	STAFPAN
Stephanus	S3152	STFNS	00182	STAFPAN

Table 2: An example of phonetic keys

3.3 Token-Based Similarity

Token-based functions divide two strings into sets of tokens s_1 and s_2, then they compute the intersection between two sets based on the number of equal tokens. Some token-based functions, for instance *Dice similarity* (DS), *Jaccard coefficient*(JS) and *Cosine similarity* (CS) (Cohen et al., 2003), consider as a token the whole word in a string. In our case most of the person names, locations and places are quite different and there are only few intersections between token-words available. Another approach, a *q-gram* (QG) tokenization (McNamee and Mayfield, 2004), divides a string into smaller tokens of size q. QG calculates the similarity between two strings by counting the number of q-grams in common and dividing by the number of q-grams in the longer string. In this paper we consider bigrams (q = 2). For example, the name 'stefan' contains the bigrams 'st', 'te', 'ef', 'fa','an'. An example of applied QG and JS similarities is shown in Table 3.

two names	LE	J	JW	SW	GH	NW	ME	QG	JS
(Stefan, Stephan)	0.71	0.85	0.89	0.58	0.57	0.79	0.57	0.5	0
(Stefan, Stephanus)	0.56	0.80	0.86	0.58	0.57	0.61	0.57	0.38	0
(Stephan, Stephanus)	0.78	0.93	0.97	1	1	0.78	1	0.75	0

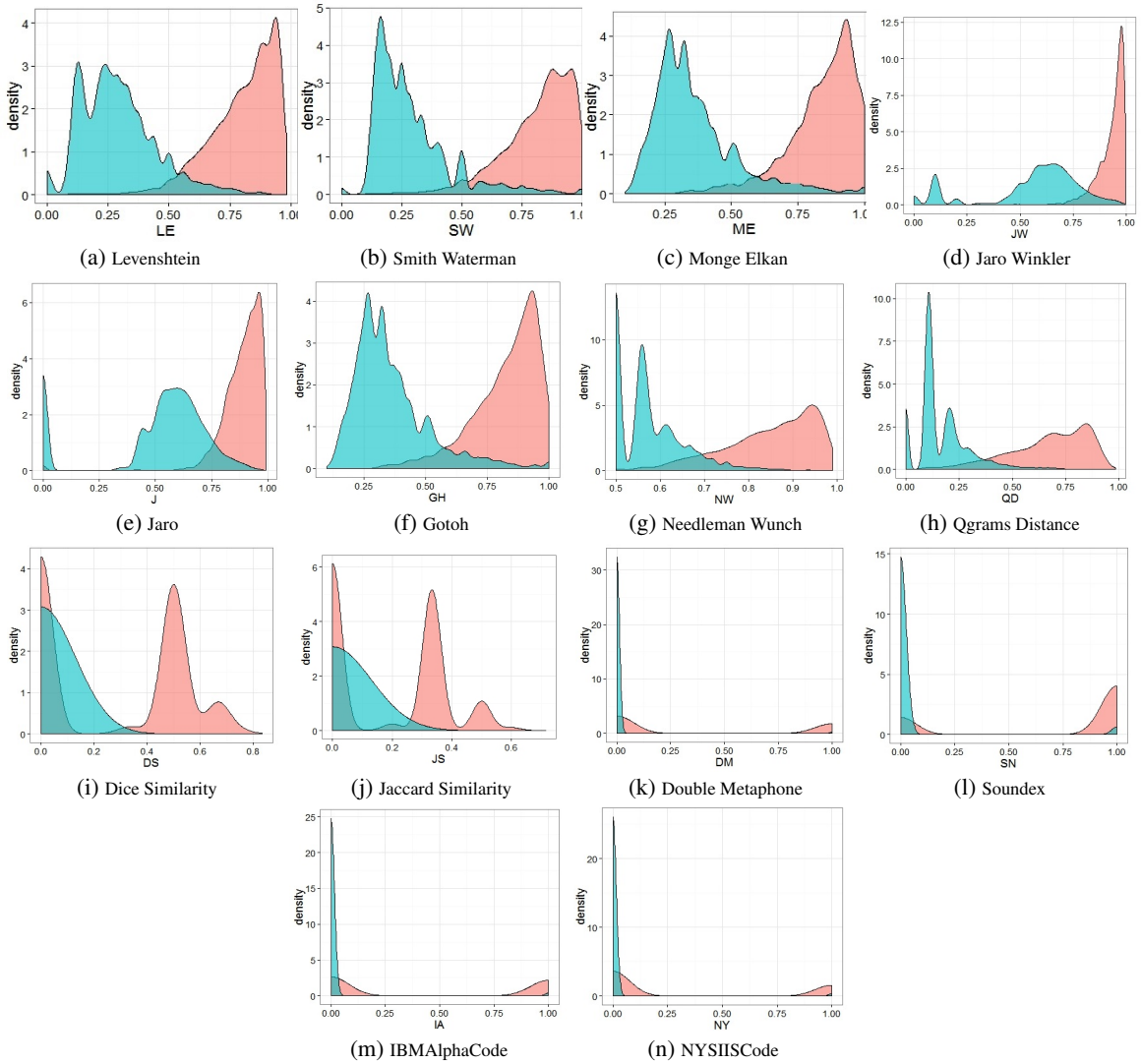
Table 3: An example of character and token-based similarities

3.4 Exploration of Standard Methods

The goal of this paper is to investigate in how far the terms variation task can be addressed by using standard methods and improve the results by applying a hybrid technique. Fig. 1 shows for each string similarity function the distribution between two non-matching pairs of records on the one hand and two matching pairs of records on the other for different measures. The more discriminative the measure is, the larger is the separation between the distributions. However, in this figure, each of similarity functions is considered independently and can be expected to only perform well in certain situations. Therefore, the goal of this paper is to design an appropriate hybrid technique, which allows to achieve better performance results by using a combination of traditional measures.

4 General Hybrid Approach

In this article we propose a new hybrid approach which takes advantage of a number of existing string similarities. Our method takes into account the most relevant string similarity by obtaining a ranking of each in terms of its importance for a classification task. The outline of the algorithm of the hybrid approach is shown below. The algorithm uses training data $\mathcal{B}$ which is provided in the form of matching and non-matching pairs of terms. First, in steps 1 to 5 the algorithm calculates pairwise similarities between two terms by every string function $(sim^1, sim^2, ..., sim^K)$. In steps 6 to 8 the algorithm computes for every sim_i an importance rate using the Random Forest technique (Genuer et al., 2010; Breiman, 2001). In subsection 4.2 we describe in more detail the process of selecting the most important string similarities. Then, in steps 10 to 22 the algorithm iteratively constructs the set of the similarity functions $\mathcal{T}^*$ which is a subset of Sim. It starts from an empty set $\mathcal{T}^*$ and at each iteration it adds to $\mathcal{T}^*$ the measure that has the highest importance rate and after that it learns the classifier $\mathcal{C}$. After every iteration the algorithm evaluates the performance



(a) Levenshtein (b) Smith Waterman (c) Monge Elkan (d) Jaro Winkler

(e) Jaro (f) Gotoh (g) Needleman Wunch (h) Qgrams Distance

(i) Dice Similarity (j) Jaccard Similarity (k) Double Metaphone (l) Soundex

(m) IBMAlphaCode (n) NYSIISCode

Figure 1: The distribution between two matching and two non-matching pairs of records for each string similarity function

in term of maximum F measure $Fmeas$ on the validation set $\mathcal{R}$. The algorithm stops if $Fmeas$ doesn't increase anymore or if the size $\mathcal{T}^*$ reaches the parameter η which can be set as a fraction of the total number of string similarity functions.

4.1 Pairwise Similarity Calculation

In order to solve the name ambiguity problem it is necessary to compute the similarity score between two records. Most of the standard string similarity functions and standard classifiers require a pairwise records comparison. We convert each dataset described in Section 2 into a dataset of variant pairs using random combinations of records. Two differently spelled terms are equal when their standardized codes are the same and different otherwise. The example of term pair-variants dataset on is shown in Table 4.

Name1	Name2	class
Statius	Eustachius	1
Statius	Stefan	0
Stefan	Stephanus	1

Table 4: An example of term pair-variants

4.2 Measure Selection

Using only the most important measures for solving the terms variation task can significantly reduce the computational cost. Therefore, we before learning the classifier we apply a selection technique. Generally, there are two common techniques that allow to reduce the number of dimensions: filters and wrappers (Das, 2001). Typically filter-based approaches require only one single scan, whereas wrapper-based ones iteratively look for the set of features which are the most suitable which leads to larger computational over-

Algorithm 1 Hybrid Disambiguation Measure

Input: Training set $\mathcal{B} = \{b_1, ..., b_\beta\}$
Validation set $\mathcal{R} = \{r_1, ..., r_\rho\}$
Set of similarity measures $Sim = (sim^1, ..., sim^K)$
Maximum allowed number of similarity measures η
$\mathcal{L}\{\mathcal{C}, \mathcal{B}, \mathcal{T}^*\}$ classifier $\mathcal{C}$ with learning algorithm $\mathcal{L}$ which is trained on the training set $\mathcal{B}$
Output: A hybrid measure Sim^{hb} based on classifier $\mathcal{C}$

```
for each b in B do
    for each sim in Sim do
        compute sim(b)
    end for
end for
for each sim in Sim do
    compute RF_sim{B}
end for
T* ← ∅
Fmeas_1{R} ← 0
i ← 2
while |T*| ≤ η do
    select sim_i that maximizes RF importance rate
    Sim ← Sim − {sim_i}
    T* ← T* ∪ {sim_i}
    L{C, B, T*}
    Calculate model performance Fmeas_i{R}
    if max(Fmeas_i{R}) > max(Fmeas_{i−1}{R}) then
        break
    end if
    i ← i + 1
end while
Sim^hb ← L{C, B, T*}
return Sim^hb corresponding to T* and C
```

heads. For designing a hybrid approach we decided to use *Random Forest* (RF) wrappers to evaluate the weight of every similarity measures. RF, according to many different sources is considered as one of the most reliable methods which is able to deal with high-dimensional and noisy data (Saeys et al., 2007). RF generates a forest of classification trees and then assign an importance rank to each similarity function based on its usefulness for the classification purpose. We use RF results to perform a *stepwise procedure* and to construct the set of measures $\mathcal{T}^*$.

4.3 Hybrid Score Computation and pairwise Classification

We consider the problem of terms variations as a prediction problem. There are many available classification techniques that are suitable for a prediction task. Many of them require a prior training phase on a representative subset of data to make a more efficient prediction on new data. After that, pairs of references are classified into classes *Matched* or *non-Matched* based on a threshold value of the score function. The score function computes the final similarity score between two terms based on results of single comparison measures. For learning the score function we use a training dataset $\mathcal{B}$. We explore 2 robust classifiers that could be applied to cultural heritage dataset domains. They are the *Logistic Regression* (LG) and the *Support Vector Machine* (SVM) (Hastie et al., 2003; Cristianini and Shawe-Taylor, 2000). They are two of the most widely-used classifiers that are suitable for the prediction task (James et al., 2013). It is important to add that we also carried out our experiments and applied three more classifiers, namely *Linear Discriminant Analysis*, *Quadratic Discriminant Analysis* and *k-nearest neighbors* (Hastie et al., 2003; Verma, 2012; Zezula et al., 2006). However results were not improved significantly on all of our datasets, so we do not include those classifiers in the designed hybrid approach.

5 The Prediction Models

In this Section we will briefly describe models that we incorporated into our hybrid approach to address the problem of inaccurate cultural heritage data.

5.1 Logistic regression

We apply a logistic regression as a predictive model and calculate the score function as follows:

$$Sim^{hb}(a_i, a_j) = \frac{1}{1 + e^{-z}}, \qquad (1)$$

where $z = \omega_0 + \omega_1 * sim^1(a_i, a_j) + \omega_2 * sim^2(a_i, a_j) + \cdots + \omega_n * sim^K(a_i, a_j)$ is an utilization of a linear regression model with parameters represented by ω_1 to ω_k. The parameters ω_0 to ω_n are learned in a training phase. The functions $sim^1(a_i, a_j)$ to $sim^K(a_i, a_j)$ represent single similarity measures between two terms a_i and a_j.

5.2 Support Vector Machines

We apply and explore SVM as a predictive model. The basic idea of SVM is that the training data is mapped into a new high dimensional space where it is possible to apply linear models to separate the classes. A kernel function performs the mapping of the training data into the new space. After that, a separation between classes is done by maximizing a separation margin between cases belonging to different classes. In our hybrid approach we use the SVM classifier with a *radial basis kernel function* and train it on the training set $\mathcal{B}$.

6 Experiments

Our experiments are conducted on four datasets. Three datasets, namely names, occupations and places variations are manually constructed from Cultural Heritage Data. They are discussed in detail in Section 2.

The fourth dataset is a public dataset called *Restaurant*. It is a standard benchmark dataset which is widely used in data matching studies (Christen, 2012; Bilenko et al., 2003). It contains information about 864 restaurant names and addresses where 112 records are duplicated. It was obtained by integrating records from two sources: Fodors and Zagats guidebooks. The *Restaurant* dataset was taken from the *SecondString* toolkit[4].

We carried out our experiments in accordance to the algorithm described in Section 4. At first, we convert each dataset into a dataset of variant pairs using random combinations of records. Then for each pair of records we compute string similarity functions. We randomly divided all available data into two subsets, namely training and test sets. To construct the set of string similarities we use 70% of the training set to learn RF importance rate and the other 30% of the training set to validate results under stepwise selection procedure as it was described in the algorithm in Section 4. The resulting set of selected string similarities for each dataset is shown in Table 5. After constructing the set of string similarities we learn the classifier on the complete training set and then evaluate it on the test set. In order to assess the performance of our results, we apply a 10-fold cross-validation method. We randomly partition the available dataset into 10 equal size subsets. Then one subset was chosen as the validation data for testing the classifier, and the remaining subsets are used for training the classifier. Then the cross-validation process is repeated 10 times, with each of the 10 subsets used exactly once as the validation dataset.

Dataset					
Names	IA	SN	DM	LE	SW
Occupations	JW	J	LE	NW	QG
Places	QG	JW	LE	SW	J
Restaurants	QG	JW	CS	NW	

Table 5: Selected string similarities during the stepwise procedure

[4]http://secondstring.sourceforge.net/

7 Evaluation Results

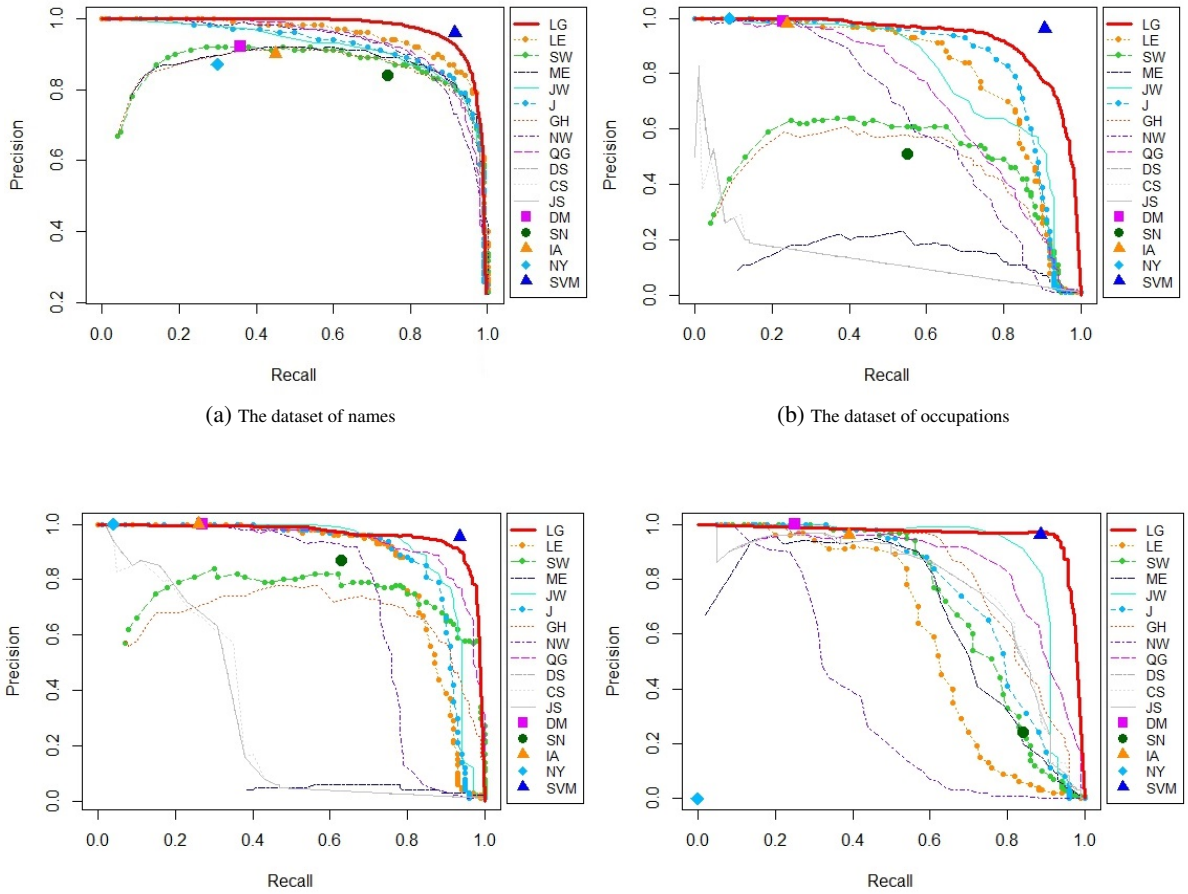
In order to evaluate the performance of standard string similarity functions and the applied hybrid approach, we compute the sets of True Positives (TP), False Positives (FP) and False Negatives (FN) as the correctly identified, incorrectly identified and incorrectly rejected matches, respectively. Fig. 2 demonstrates the performance of standard and hybrid approaches on four examined datasets.

The logistic regression as well as SVM classifiers which are used in the hybrid approach on each of the dataset outperform standard string similarities. The improvement in results is significant, especially it is clearly seen on the dataset of occupations. For a more detailed analysis, Fig. 3 shows the evaluation of results in terms of F-measure and the threshold value for all continuous methods. Moreover, Table 6 shows the maximum values of the F-measure for the five best performing methods for each of the datasets. Two upper rows of the table belong only to the hybrid approach. SVM and logistic regression in the combination with the RF selection technique both demonstrate robustness on the multiple datasets domains.

Names		Occupations		Places		Restaurants	
Method	Max.F	Method	Max.F	Method	Max.F	Method	Max.F
SVM	0.94	SVM	0.93	SVM	0.95	LG	0.95
LG	0.92	LG	0.86	LG	0.93	SVM	0.91
LE	0.89	J	0.82	QG	0.88	JW	0.87
J	0.87	LE	0.77	JW	0.87	QG	0.81
SW	0.86	JW	0.71	J	0.85	GH	0.76

Table 6: The maximum F-Measure values for the five best-performing methods

In addition to analyzing the hybrid approach, in this section we investigate in more detail functions which demonstrate not the typical behavior on the precision and recall plots. For instance, on Fig. 2 for SW, GH and ME similarities the simultaneous growth of the precision is accomplished by the growth in the recall on the interval (0, 0.3). The same situation occurs for SW and GH similarities on datasets of occupations and places. In Table 7 for SW similarity on the dataset on names for three levels of the threshold we show its performance indicators, namely TP, FP and FN values and calculated the precision and recall. With the maximum similarity score SW incorrectly identify as positive 99 pairs of names. With the slightly decrease in the threshold value, the larger number of pairs are identified correctly as the variation of



(a) The dataset of names

(b) The dataset of occupations

(c) The dataset of places

(d) Public dataset of restaurants

Figure 2: Evaluation of single string similarities and the hybrid approach in terms of precision and recall

Threshold	TP	FP	FN	Precision	Recall
0.96	809	99	3853	0.89	0.17
0.98	355	99	4307	0.78	0.08
1	200	99	4462	0.67	0.04

Table 7: Evaluation measures for SW similarity for 3 levels of the threshold

Name1	Name2	SW	GH	ME
Peternella	Peter	1	1	1
Pauline	Paul	1	1	1
Henriette	Henri	1	1	1

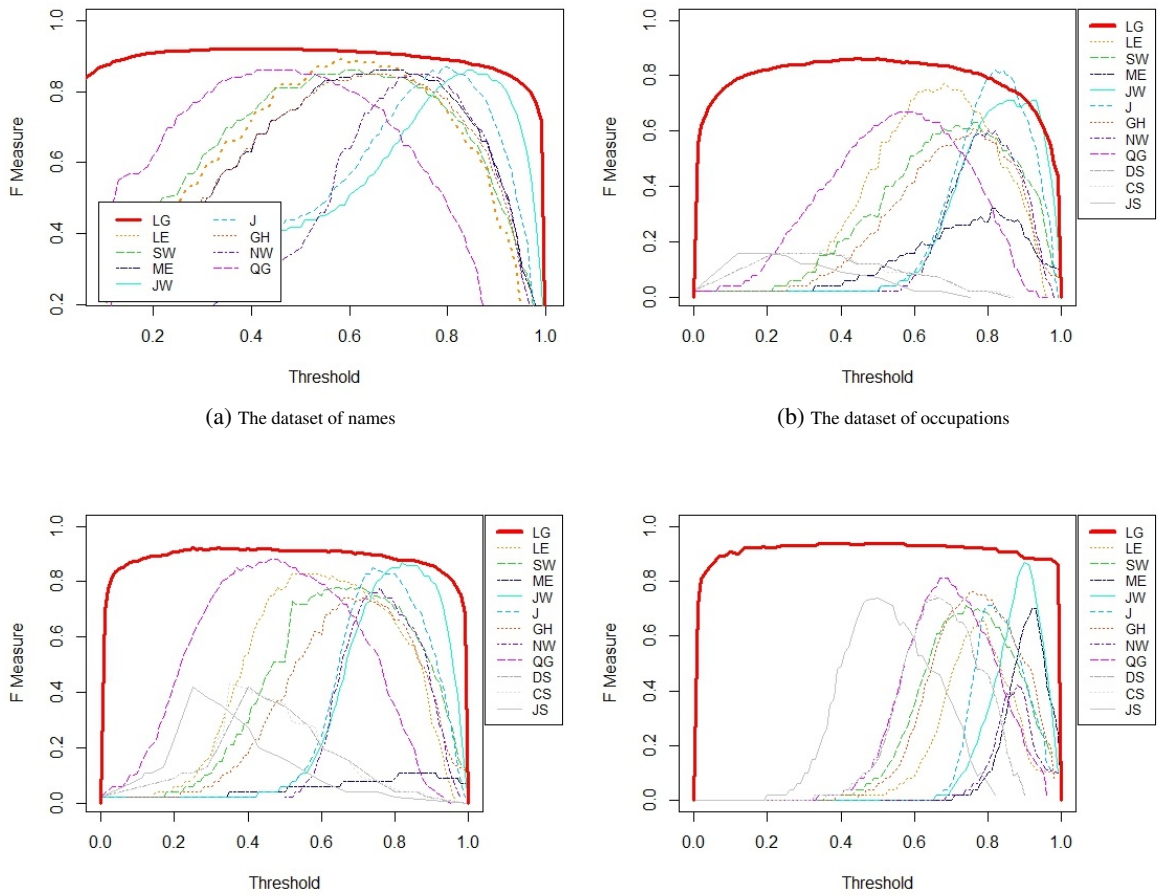
Table 8: Example of FP pairs of names according to the maximum value of SW, GH and ME functions

the same name. Therefore, to make it absolutely clear, in Table 8 we gave an example of such pairs of names that are included into 99 FP and cause the simultaneous grows of precision and recall.

8 Discussion

The proposed hybrid approach shows very good results in performing the pairwise terms comparison for completely different dataset domains. Nevertheless, the bottleneck of the algorithm is that it is expensive to apply it to real-world data and compare all possible combinations of records.

There are various available techniques for reducing the amount of candidate pairs to be compared. Common techniques are partitioning data into smaller subsets and comparing only records with the same partition. Two widely used partition approaches are blocking and windowing methods (Naumann and Herschel, 2010; Bilenko et al., 2003). The blocking technique assigns to each record a special blocking key, for instance year, place of the documents or the first 3 letters of the last name. The windowing technique such as



(a) The dataset of names

(b) The dataset of occupations

(c) The dataset of places

(d) Public dataset of restaurants

Figure 3: Evaluation of single string similarities and the hybrid approach in terms of F-Measure and Threshold

Sorted Neighborhood method sorts data according to some key, for instance year of the documents, and then slides a window of fixed size across the sorted data. Reducing the number of candidate pairs may result that two references that refer to the same entity appear in different partitions and then they will never be compared. Therefore, our next work will focus on searching the best partition method (or best hybrid methods) that allows to reduce the number of potential candidate pairs and keep all references referring to the same entity within the same partition.

9 Conclusion

In this paper we studied a number of traditional string similarities and proposed the hybrid approach applied on different cultural heritage dataset domains. It is obvious that dealing with historical documents, where attributes information is often imprecise, is not possible by using only one string similarity. Therefore, we investigated how to improve the performance by using a number of string similarities and applied supervised learning technique.

As future step, the authors are working on incorporating the hybrid approach into overall entity resolution process in a large genealogical database (Efremova et al., 2014), which aims to discover which of the person references mentioned in different historical documents refer to the same person entity. The genealogical database contains a collection of historical documents where names, occupations and places are the essential attributes. Therefore it is very important to find a robust and reliable approach which is able to compare main personal information in the noisy data.

10 Acknowledgments

The authors are grateful to the BHIC Center, in particular to Rien Wols and Anton Schuttelaars for the support in data gathering, data analysis and direction.

References

Indrajit Bhattacharya and Lise Getoor. 2007. Collective entity resolution in relational data. *ACM Trans. Knowl. Discov. Data*, 1(1).

Mikhail Bilenko and Raymond J. Mooney. 2003. Adaptive duplicate detection using learnable string similarity measures. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '03, pages 39–48. ACM.

Mikhail Bilenko, Raymond Mooney, William Cohen, Pradeep Ravikumar, and Stephen Fienberg. 2003. Adaptive name matching in information integration. *Intelligent Systems*, 18(5):16–23.

Leo Breiman. 2001. Random forests. *Machine Learning*, 45(1):5–32.

Peter Christen. 2006. A comparison of personal name matching: techniques and practical issues. In *Proceedings of the Workshop on Mining Complex Data (MCD06), held at IEEE ICDM06*, pages 290–294.

Peter Christen. 2012. *Data matching*. Springer Publishing Company, Incorporated.

William W. Cohen, Pradeep Ravikumar, and Stephen E. Fienberg. 2003. A comparison of string distance metrics for name-matching tasks. In *Proceedings of IJCAI-03 Workshop on Information Integration*, pages 73–78.

Nello Cristianini and John Shawe-Taylor. 2000. *An introduction to Support Vector Machines: and other kernel-based learning methods*. Cambridge University Press.

Sanmay Das. 2001. Filters, wrappers and a boosting-based hybrid for feature selection. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 74–81. Morgan Kaufmann Publishers Inc.

Julia Efremova, Bijan Ranjbar-Sahraei, Frans A. Oliehoek, Toon Calders, and Karl Tuyls. 2014. A baseline method for genealogical entity resolution. In *Proceedings of the Workshop on Population Reconstruction, organized in the framework of the LINKS project*.

Ahmed K. Elmagarmid, Panagiotis G. Ipeirotis, and Vassilios S. Verykios. 2007. Duplicate record detection: A survey. *Knowledge and Data Engineering, IEEE Transactions on*, 19(1):1–16.

Robin Genuer, Jean-Michel Poggi, and Christine Tuleau-Malot. 2010. Variable selection using random forests. *Pattern Recognition Letters*, 31(14):2225–2236.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. 2003. *The elements of statistical learning*. Springer, corrected edition.

Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. 2013. *An introduction to statistical learning: with applications in R*. Springer Publishing Company, Incorporated.

Bertrand Lisbach and Victoria Meyer. 2013. *Linguistic identity matching*. Springer.

Kees Mandemakers, Sanne Muurling, Ineke Maas, Bart Van de Putte, Richard L. Zijdeman, Paul Lambert, Marco H.D. van Leeuwen, Frans van Poppel, and Andrew Miles. 2013. *HSN standardized, HISCO-coded and classified occupational titles*. IISG Amsterdam.

Paul McNamee and James Mayfield. 2004. Character n-gram tokenization for european language text retrieval. *Information Retrieval*, 7(1-2):73–97.

Felix Naumann and Melanie Herschel. 2010. *An Introduction to Duplicate Detection*. Morgan and Claypool Publishers.

Gonzalo Navarro. 2001. A guided tour to approximate string matching. *ACM Comput. Surv.*, 33(1):31–88.

Eric Sven Ristad, Peter N. Yianilos, and Senior Member. 1998. Learning string edit distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20:522–532.

Yvan Saeys, Iñaki Inza, and Pedro Larrañaga. 2007. A review of feature selection techniques in bioinformatics. *Bioinformatics*, 23(19):2507–2517, September.

Sheila Tejada, Craig A. Knoblock, and Steven Minton. 2002. Learning domain-independent string transformation weights for high accuracy object identification. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '02, pages 350–359. ACM.

Marco H. D. van Leeuwen, Ineke Maas, and Andrew Miles. 2002. *HISCO. Historical international standard classification of occupations*. Leuven University Press.

J P Verma. 2012. *Data Analysis in Management with SPSS Software*. Springer.

William E. Winkler. 1995. Matching and record linkage. In *Business Survey Methods*, pages 355–384. Wiley.

Pavel Zezula, Giuseppe Amato, Vlastislav Dohnal, and Michal Batko. 2006. *Similarity search: the metric space approach*. Springer.