Who Are my Ancestors? Retrieving Family Relationships from Historical Texts

Julia Efremova[1], Alejandro Montes García[1], Alfredo Bolt Iriondo[1], and Toon Calders[1,2]

[1] Eindhoven University of Technology, The Netherlands {i.efremova, a.montes.garcia, a.bolt.iriondo}@tue.nl
[2] Université Libre de Bruxelles, Belgium toon.calders@ulb.ac.be

Abstract. This paper presents an approach for automatically retrieving family relationships from a real-world collection of Dutch historical notary acts. We aim to retrieve relationships like *husband - wife*, *parent - child*, *widow of*, etc. Our approach includes person names extraction, reference disambiguation, candidate generation and family relationship prediction. Since we have a limited amount of training data, we evaluate different feature configurations based on the n-gram analysis. The best results were obtained by using a combination of bi-grams and tri-grams of words together with the distance in words between two names. We evaluate our results for each type of the relationships in terms of precision, recall and $f - score$.

1 Introduction

Extraction of characteristics from the text is one of the main tasks in text mining. Structured information retrieved from the text can be used for different purposes, for instance, documents classification, filtering emails, finding key words, etc. Having extracted person names and their relationships, makes available a large amount of personal information. Previous research showed good results in automated extraction of skills from job applicants to make job application process more efficient [10].

Family relationships is a special type of person relationships. It is an important step in linking persons across different genealogical documents and sources [7,6]. As an example, consider a couple *Martinus de Jager* and *Hendrina Jacobs* who married in *1888*. The information about their marriage is recorded in a civil register. Two years later *Martinus* and *Hendrina* are mentioned as husband and wife in a notary act because they bought a house. Having extracted the *husband-wife* relationship between them can help to link this notary act to the marriage certificate of the mentioned couple or the birth certificates of their children where *Martinus* and *Hendrina* are mentioned as parents.

As such, extraction of family relationships from text documents is important for population reconstruction which is a key element in the genealogical research and social studies. Family relationships extraction can be also used in text classification. For instance, regarding a collection of notary acts, it is much easier to

distinguish between an inheritance act and a purchase activity because the first type contains many family relationships and the second usually not.

In this paper we focus on the extraction of family relationships (FR) from historical notary acts which is a challenging text mining problem. We deal with identification of FRs objects, FRs key words abbreviations, name variation in the text or implicit relationships. We identify main components such as extraction of named entities (person names in our case), name disambiguation if a person is mentioned more than once in the document, and relationship prediction.

Our contributions can be summarized as follows:

- We propose a framework that allows the retrieval of family relationships extraction from the text with minimal available training data.
- We report results of n-gram analysis that are used as a feature configuration technique.
- We provide a training collection that consists of labeled notary acts and relationships pairs to the research community.

The remainder of this paper has the following structure. In Section 2 we discuss related work. In Section 3 we describe the data collection. In Section 4 we present a general process of family relationship extraction. In Section 5 we set up the experiments and present the results. Finally, we make a conclusion in Section 6.

2 Related Work

In this section we discuss the related work on family relationships extraction. Santos et al. [15] apply a rule-based approach in order to extract family relationships. This method contains of 99 different rules which allow to identify and to classify family relationships. Their obtained f-score varies from 29% to 36% and the designed rule based patterns are restricted to the Spanish text. Makazhanov [12] extracts FR networks from literary novels. He uses literature narratives and considers utterances in the text which are attributed into different categories: quotes, apparent conversations, character tri-gram and others. Then the FR prediction is done by using a Naive Bayes classifier and it is evaluated on the book of Jane Austen *Pride and Prejudice*. Kokkinatis and Malm [11] describe an unsupervised approach to extract interpersonal relations between identified person entities from Swedish prose. Recently Collovini et al. [3] designed a process for the extraction of any types of relations between named entities for Portuguese text in the domain of organisations. They apply statistical modelling with different feature combinations. Bird et al. [2] describe relationship extraction based on regular expressions and pattern features. However, their method requires a dictionary of named entities. For instance, they use *in* pattern to find the location of organisations: *[ORG: Bastille Opera] 'in' [LOC: Paris]*. Mintz et al. [13] propose an approach for relation extraction from the text that does not require labelled data. They focus on identifying pairs, for example, the *person-nationality* relation which holds between person entities and nationality entities. In our work we aim to identify triples *(person$_1$, family relationship, person$_2$)*. Based on the

previous work applied to different languages and application domains we design our own framework for FR extraction from historical documents.

We also mention a number of general work available in text classification. One of the main surveys in text classification (TC) was published by Sebastiani [16] where the author described main TC steps such as document indexing, dimensionality reduction, key term selection, learning a classifier and evaluation. Ikomomakis [9] extended later his work and made a summary of the available TC technique which contain a number of relevant cited references. One of the recent survey belong to Aggarwal and Zhai [1]. They described main TC components together with state-of-the-art solutions from data mining, machine learning and information retrieval.

3 Data Description

The dataset used is a subset of the one described in our previous work [5] and it is available on the web [3]. More specifically we give the description of the annotated input collection in Section 5.1. The collection contains historical notary acts such as: property transfer, sale, inheritance, public sale, obligation, declaration, partition of inheritance, resolution, inventory and evaluation for a period of around 500 years. Many of the notary acts contain information about people and family relationships between them. Since the documents belong to a time period that was many years ago sometimes they are the only sources of information regarding the population and family relations of that period. Thus, we need an efficient technique to extract person entities and their relationships.

Below is an example of a notary act that has the *husband-wife* relationship (the person names are underlined and relationships are in bold):

> Dit document certificeert: Martinus de Jager en **zijn vrouw** Hendrina Jacobs, verklaren afstand te doen van alle rechten van de akte van koop en verkoop van 02/10/1906, opgemaakt voor notaris van Breda, ten behoeve van Martinus van Doorn, winkelier te Uden.
> *This document certifies: Martinus de Jager and* ***his wife*** *Hendrina Jacobs, declare to waive all rights of the act of sale and purchase of 02/10/1906, registered at the notary Breda, as beneficiary Martinus van Doorn, shopkeeper in Uden.*

The average length of the documents is 70 words, although there exist some documents with up to 1,000 words. The overall collection contains around 115,000 notary acts with dates ranging from *1433* to *1920*. However, the majority of documents belongs to the period *1650-1850*. Different time period of documents and

[3] urlhttp://goo.gl/leibR9

different documenting standards make the task of family relationship extraction very difficult.

4 Family Relationship Extraction

In this section we discuss the following steps of the FR retrieval process: *raw data pre-processing*, *name extraction*, *name disambiguation*, *candidate pair generation*, *feature generation* and *classification*.

4.1 Name Extraction

To extract person names from notary acts we use a collection of Dutch first and last names obtained from the website of Meertens Institute[4] available in Dutch only. It contains around 115,000 different last names, 18,000 male and 26,000 female first names. We use this database as a name dictionary. Although the name dictionary is large, we can not apply it directly and tag all first and last names in the text. Some name variations might be missed. To avoid this situations we designed our own name extraction that proceeds in three steps.

In the first step, we define a set of labels {*FN, LN, I, P, CAP, O*} in which *'FN'* and *'LN'* stand for first and last names respectively, the tag *'I'* refers to a name initial (one letter followed by a dot like *'W.'* instead of *'Willem'*), *'P'* is a name prefix like *van, der, de*, *'CAP'* corresponds to other words that start from a capital letter and *'O'* indicates that there is no name descriptor. We assign an appropriate label to every word in the document in two iterations. We first tag first names and last names using the name dictionary, then we tag initials, name prefixes, words that start from a capital letter and other words that are not tagged yet.

In the second step we design name patterns using regular expressions. The phrase in the text is extracted as a name if it meets the requirements of a name pattern. Table 1 shows the three main name patterns that we used to specify a name phrase. The first name pattern corresponds to the situation when at least one first name exists in the dictionary. A last name is optional in this case and can be tagged as *'LN'* or *'CAP'*. If the last name does not exist in the dictionary we consider a word after the first name that starts from a capital letter as the last name. Between first and last name, initials or a name prefix may appear. This rule allows us to extract a single first name and full names at the same time. The second expression in Table 1 finds names that start from initials followed by the last name which can be tagged again as *'LN'* or *'CAP'*. The third expression requires a last name tag whereas the first name can be labelled with *'FN'* or *'CAP'* tags. We illustrate the process of labelling words and finding name patterns in Fig. 1.

In the third step we make name disambiguation and merge the same names into one. Name disambiguation is a necessary step in case when a person is

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

Table 1: The grammar that specifies a name pattern

No.	Name pattern
1	{<CAP>?<FN>+<I>?<P>?<LN|CAP>?}
2	{(<I>)+<FN>?<I>?<LN|CAP>+}
3	{<FN|CAP>+<P>?+<LN>}

mentioned multiple times. However, it is not a fully person resolution methods. The goal of this paper is to identify pair of persons in a documents and predict their relationships. In order to do it we make a pairwise relationship prediction between two names. The extraction of person entities is not in the scope of this paper and is considered as a part of future work. The problem is that different persons can have the same name, for instance *Hendrina Jacobs and her daughter Hendrina Jacobs* and the same person can be mentioned differently: *Hendrina Jacob* and *H. Jacobs*. Our hypothesis is that family extraction technique can be a component of person resolution process. That is why we deal with names but not with different persons.

This simple technique extracts names with high accuracy, efficiently deals with abbreviations in them: *W. G. van Oijen* or *Jan J. Beckers* and distinguishes person names from other information and location in the text. For instance, compare the name *Jan van Erp* and the phrase *Kerk van Erp*[5]. The proposed name extraction technique is able to distinguish these two situations from each other.

4.2 General Approach

To retrieve FRs we start with data preprocessing and remove from the raw data all non-alphabetical symbols except punctuation marks. In the next step we extract person names from notary acts and perform name disambiguation as described in Section 4.1. For every pair of names extracted from the same notary act, we construct a candidate pair and create a feature vector for that pair. The feature vectors are constructed as follows. We consider all words between the two names in a pair and also two words before the first name and two words after the last name as illustrated in Fig. 1. Thus, for each candidate pair we identify a set of words called *tokens*.

We compute *term frequency* for each token in a candidate pair. The output of the feature extraction step is a set of numerical features. The created vocabulary is large. We experiment with different sets of features: *bi-grams* of words, *tri-grams* of words, a combination of *bi-grams* and *tri-grams*. We also add the length in words between two names and consider two situations: with and without length information.

[5] 'Kerk van Erp' in Dutch means 'church of Erp'

The last step of the FR process is learning the model and classifying candidate pairs into FR type or *No-FR*. We apply and evaluate the designed technique using two classifiers: a linear *Support Vector Machine* (SVM) and multinomial *Naive Bayes* [8,14] from the scikit-learn python tool[6]. In our previous work in [5] we investigated a number of other predictive models such as *Ridge regression, Perceptron, Passive Aggressive, Stochastic gradient descent, Nearest centroid* applied for document classification tasks [16]. The results are available on the web [7]. We choose SVM and Naive Bayes classifiers because SVM classifier showed the highest performance results in our previous study and Naive Bayes classifier is typically considered as a baseline in text classification [1].

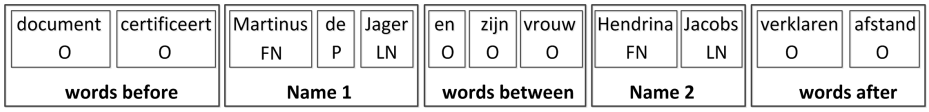


Fig. 1: The illustration of tagging words in a notary act, name extraction and creation of a feature vector for a candidate pair

5 Experiments

The extraction of family relationships from notary acts and its evaluation require additional steps. The first step is the process of gathering expert opinions. This is a crucial requirement for the evaluation and training a prediction model. Therefore in this section first we present an interactive web-based interface which was used for getting input from humans. Then we elaborate on the application and the evaluation of the model. We apply 10-fold cross-validation to evaluate our method.

5.1 Notary Act Annotation

Fig. 2 presents a screenshot of a developed web application for indexing family relationships. We asked experts to index notary acts and manually extract family relationships such as *parent of, siblings, married to, widow of, etc.* Experts perform pairwise data annotation via the interface. First, they identify two person names that have a relationship then they specify the relationship type. Using the developed tool the experts manually annotated *1,005* family relationships that belong to *347* annotated notary acts. The distribution between the different types of family relationship is provided in Table 2. It is very costly to obtain labeled data. Therefore, we need a technique which is able to learn a model using a minimal number of training examples.

[6] `http://scikit-learn.org/`
[7] `http://wwwis.win.tue.nl/amontes/ecir2015/results.html`

Table 2: Statistics of manual annotation

	Marriage	Parent-child	Widow of	Sibling to	Nephew of
Number of relationships	530	298	121	45	11
Number of different relationship descriptors	43	35	21	17	4

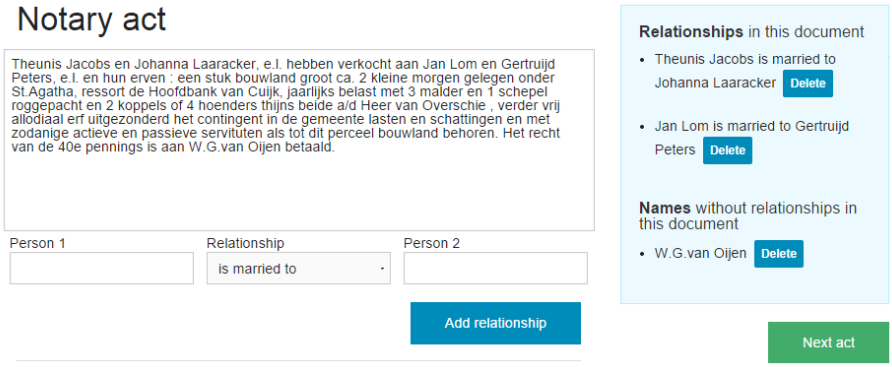
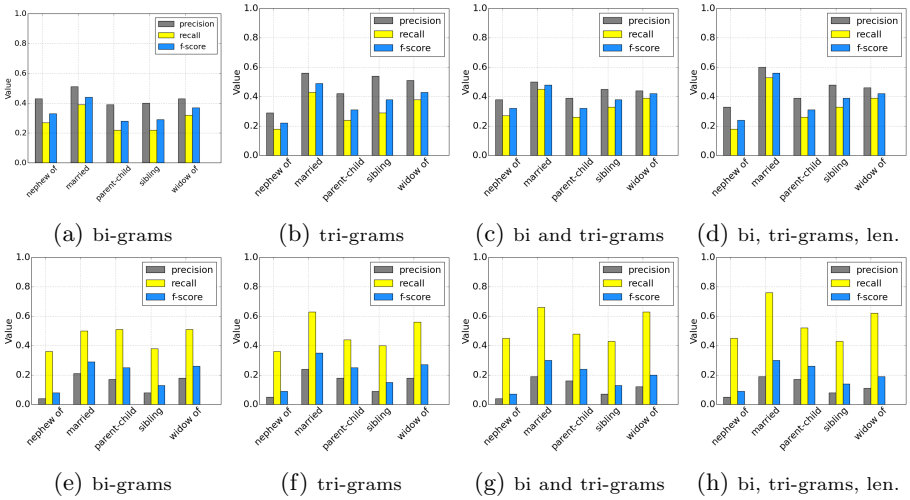


Fig. 2: The designed web-based interface for annotating person names and family relationships

5.2 Result Evaluation

We evaluate the performance of the applied algorithms in terms of precision, recall, and F-score. Fig. 3 show the results for different feature configurations and two classifiers: a Support Vector Machine (Fig. 3a-3d) and Naive Bayes (Fig. 3e-3h). The maximum *f-score* we achieve for marriage relationships using the SVM classifier and a combination of bi-grams, tri-grams of words and length between two names as presented in Fig. 3d. Marriage relationships are the most frequent among other FR types, and as such more training examples are available. Another reason is that *marriage* relationship is explicit and it is clearly mentioned in the text, in contrast to *parent-child* and *siblings*. The last two types might require an additional propagation. For instance, if a mother and her two kids are mentioned in the text, then these two children are siblings of each other. In this case we first need to predict correctly parent-child links and then retrieve sibling relationship for parents that have more than one kid. The relationships *widow of* are also explicit relationships and the classifier recognizes them with the *f-score* above 0.4 despite of very small number of training examples. Overall, the SVM classifier outperforms Naive Bayes. We see that combining features together improves the SVM classification.



(a) bi-grams (b) tri-grams (c) bi and tri-grams (d) bi, tri-grams, len.

(e) bi-grams (f) tri-grams (g) bi and tri-grams (h) bi, tri-grams, len.

Fig. 3: Comparison of performance results for different feature configurations: (a)-(d) after applying the SVM classifier, (e)-(h) after applying Naive Bayes classifier. Len. stands for length

6 Conclusions

In this paper we introduced a framework for family relationship extraction from historical notary acts. We examined different feature combinations and presented the initial results produced by two machine learning classifiers. The performance varies for different types of relationships; however we identified many family relationships correctly. We missed some family relationships because not all of them are explicitly mentioned in the text, especially concerning parent-child and siblings relationships. As a future work we plan to retrieve an implicit family relationship that require initial prediction. Another extension concerns the predictive model, where we plan to explore the use of Hidden Markov Models [4] which is widely for the text annotation purposes. However text annotation task is very different from the task of family extraction and require a number of steps in order to convert the annotated corpora into the pair of names with a specified relationship.

Acknowledgements

Mining Social Structures from Genealogical Data (project no. 640.005.003) project, part of the CATCH program funded by the Netherlands Organization for Scientific Research (NWO).

References

1. Charu C. Aggarwal and ChengXiang Zhai. A survey of text classification algorithms. In *Mining Text Data*, pages 163–222. Springer, 2012.
2. Steven Bird, Ewan Klein, and Edward Loper. *Natural Language Processing with Python*. O'Reilly Media, Inc., 1st edition, 2009.
3. Sandra Collovini, Lucas Pugens, Aline A. Vanin, and Renata Vieira. Extraction of relation descriptors for portuguese using conditional random fields. In *Advances in Artificial Intelligence - IBERAMIA 2014 - 14th Ibero-American Conference on AI, Santiago de Chile, Chile, November 24-27, 2014, Proceedings*, pages 108–119, 2014.
4. Sean R Eddy. What is a hidden markov model? *Nat Biotech*, 22(10):1315–1316, October 2004.
5. Julia Efremova, Alejandro Montes García, and Toon Calders. Classification of historical notary acts with noisy labels. In *In Proceedings of the 37th European Conference on Information Retrieval*, ECIR'15, Vienna, Austria, 2015. Springer.
6. Julia Efremova, Bijan Ranjbar-Sahraei, Frans A. Oliehoek, Toon Calders, and Karl Tuyls. An interactive, web-based tool for genealogical entity resolution. In *25th Benelux Conference on Artificial Intelligence(BNAIC'13)*, The Netherlands, 2013.
7. Julia Efremova, Bijan Ranjbar-Sahraei, Frans A. Oliehoek, Toon Calders, and Karl Tuyls. A baseline method for genealogical entity resolution. In *Proceedings of the Workshop on Population Reconstruction, organized in the framework of the LINKS project*, 2014.
8. Eibe Frank and Remco R. Bouckaert. Naive bayes for text classification with unbalanced classes. In *PKDD*, pages 503–510, Berlin, Heidelberg, 2006. Springer-Verlag.
9. M. Ikonomakis, S. Kotsiantis, and V. Tampakas. Text classification using machine learning techniques, 2005.
10. Ilkka Kivimki, Alexander Panchenko, Adrien Dessy, Dries Verdegem, Pascal Francq, Cdrick Fairon, Hugues Bersini, and Marco Saerens. A graph-based approach to skill extraction from text, 2013.
11. Dimitrios Kokkinakis and Mats Malm. Character profiling in 19th century fiction, 2011.
12. Aibek Makazhanov, Denilson Barbosa, and Grzegorz Kondrak. Extracting family relationship networks from novels. *CoRR*, 2014.
13. Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2 - Volume 2*, ACL '09, pages 1003–1011, USA, 2009. Association for Computational Linguistics.
14. Geoffrey I. Sammut, Claude; Webb. *Encyclopedia of Machine Learning*. Springer, Berlin Heidelberg, 2010.
15. Daniel Santos, Nuno Mamede, and Jorge Baptista. Extraction of family relations between entities. In *INForum 2010: - II Simpsio de Informtica*, 2010.
16. Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM Comput. Surv.*, 34(1):1–47, 2002.