ECAI 2010

Proceedings of the ECAI 2010 Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities (LaTeCH 2010)

16 August, 2010
Faculty of Science, University of Lisbon
Lisbon, Portugal

Preface

The LaTeCH (*Language Technology for Cultural Heritage, Social Sciences, and Humanities*) workshop series aims to provide a forum for researchers who are working on developing novel information technology for improved information access to data from the Humanities, Social Sciences, and Cultural Heritage. The first LaTeCH workshop was held in 2007, as a satellite workshop at ACL (*Annual Meeting of the Association for Computational Linguistics*), in Prague. There have been since three further editions of the workshop: at LREC 2008 (*Language Resources and Evaluation Conference*), in Marrakech, at EACL 2009 (*Conference of the European Chapter of the Association for Computational Linguistics*), in Athens, and now at ECAI 2010, in Lisbon. While the initial focus was on 'Cultural Heritage', it has gradually broadened to also include the Humanities and the Social Sciences. All three areas have in common that language data, i.e., text and –to a lesser extent– speech, play a central role, both as primary and secondary data sources. Current developments in these areas have resulted in an increased amount of data becoming available in electronic format, either as the outcome of recent digitisation efforts, or in the form of born-digital data. What is often lacking, nevertheless, is the technology to process and access these data in an intelligent way. Information technology research and applications can provide solutions to this problem, such as methods for data cleaning and data enrichment with semantic information, so as to support more sophisticated querying, and discovery and visualisation of interesting data trends. While the Humanities, Social Sciences and Cultural Heritage domains clearly benefit from this type of research, these domains also provide a challenging test bed for information technology. Traditionally, language information technology has been focused on other domains, such as newswire. Data from the Humanities, Social Sciences and Cultural Heritage entail new challenges, such as noisy text (e.g., due to OCR problems), non-standard, or archaic language varieties, the necessity to link data of diverse formats (e.g., text, database, video, speech) and languages, and the lack of large annotated data sets for supervised machine learning solutions. Researchers consequently have to be creative in developing robust methods for these domains.

While the main focus of LaTeCH is on language technology, for the current edition of the workshop we broadened the scope and invited papers from related areas, including machine learning, pattern recognition, knowledge representation, multi-modal systems, recommender systems, and neighbouring fields in AI. Papers were accepted for LaTeCH 2010 after a thorough peer-review process and the selected papers give a good overview of the current breadth of this exciting and expanding area. On the technology side the papers cover topics ranging from preprocessing and error detection, over semantic annotation and information extraction to data visualisation. The contributions also deal with a wide variety of domains, including folk tales and ritual descriptions, cabinet minutes and political speeches, letters, legal documents, Hungarian codices, Alpine literature and audio-video streams.

We would like to thank all authors who submitted papers for the hard work that went into their submissions. We are also extremely grateful to the members of the programme committee for their thorough reviews, and to the ECAI 2010 workshop chair, Ulle Endriss, for help with administrative matters.

Caroline Sporleder & Kalliopi Zervanou

LaTeCH 2010 Organisation

Programme Chairs

Caroline Sporleder, *Saarland University* (Germany)
Kalliopi Zervanou, *University of Tilburg* (The Netherlands)

Organising Committee

Caroline Sporleder, *Saarland University* (Germany)
Kalliopi Zervanou, *University of Tilburg* (The Netherlands)
Lars Borin, *Göteborgs Universitet* (Sweden)
Piroska Lendvai, *Academy of Sciences* (Hungary)
Antal van den Bosch, *University of Tilburg* (The Netherlands)

Programme Committee

Ion Androutsopoulos, *Athens University of Economics and Business* (Greece),
Tim Baldwin, *University of Melbourne* (Australia)
David Bamman, *Tufts University* (USA)
Toine Bogers, *Royal School of Library and Information Science* (Denmark)
Lars Borin, *Göteborgs Universitet* (Sweden)
Antal van den Bosch, *University of Tilburg* (The Netherlands)
Paul Buitelaar, *DERI Galway* (Ireland)
Kate Byrne, *University of Edinburgh* (Scotland)
Milena Dobreva, *HATII, University of Glasgow* (Scotland)
Mick O'Donnell, *Universidad Autonoma de Madrid* (Spain)
Julio Gonzalo, *Universidad Nacional de Educacion a Distancia* (Spain)
Claire Grover, *University of Edinburgh* (Scotland)
Ben Hachey, *Macquarie University* (Australia)
Dominik Heckmann, *DFKI* (Germany)
Christer Johansson, *University of Bergen* (Norway)
Jaap Kamps, *Universiteit van Amsterdam* (The Netherlands)
Vangelis Karkaletsis, *NCSR "Demokritos"* (Greece)
Michael Kipp, *DFKI* (Germany)
Stasinos Konstantopoulos, *NCSR "Demokritos"* (Greece)
Piroska Lendvai, *Academy of Sciences* (Hungary)
Véronique Malaisé, *Vrije Universiteit Amsterdam* (The Netherlands)
Barbara McGillivray, *Università degli Studi di Pisa* (Italy)
John McNaught, *University of Manchester, NaCTeM* (UK)
Ruslan Mitkov, *University of Wolverhampton* (UK)
John Nerbonne, *Rijksuniversiteit Groningen* (The Netherlands)
Katerina Pastra, *Institute for Language and Speech Processing* (Greece)
Marco Pennacchiotti, *Yahoo! Research* (USA)
Georg Rehm, *DFKI* (Germany)
Martin Reynaert, *University of Tilburg* (The Netherlands)
Svitlana Zinger, *TU Eindhoven* (The Netherlands)

Table of Contents

Usability Enhancement by Mining, Processing and Visualizing Data from the Federal German Archive

Andreas Schwarte and **Christopher Haccius** and **Sebastian Steenbuck** and **Sven Steudter** [1]

Abstract. The purpose of this paper is to present the results of a project which deals with mining data from historical corpora. We present a general approach of how existing language processing tools can be used and integrated to automate information retrieval and data processing. Moreover, we discuss ideas of flexible access to the data as well as presentation. Our findings and results are illustrated in a prototype system which allows for multi-dimensional queries on the corpus. The corpus for this project consists of the German cabinet meeting protocols from 1949 to 1964.

1 INTRODUCTION

The study of history usually requires the extraction of relevant data out of large corpora. This data extraction is typically very time-consuming if done manually; time which is afterwards lacking for the actual research. It is therefore urgently necessary to make data on the one hand accessible, and on the other hand automate the process of data extraction and presentation. Several tools have already been developed to facilitate language processing and data analysis. Many of these are available as libraries for further use.

The main idea of this project is to use existing data processing methods and combine them to produce a result which can be used for further research. The ultimate project goal is to retrieve information from a given historical data source, to convert unstructured data into a searchable database, and most important enable the discovery of new knowledge as well as links between documents in the corpus. We discuss techniques for topic identification, present methods and data structures that allow efficient searching, and give ideas about data presentation and visualization. The purpose of this paper is to elaborate on a general approach of how the different methodologies can be integrated into a system that is able to facilitate information processing of large data sources.

While several different data sources are applicable, we present a prototype dealing with the meeting protocols of the German federal cabinet from 1949 to 1964[2]. On the basis of this corpus we present a procedure and implementation how to tackle large data sources by computational means. The system presented below is able to retrieve raw data from the given source, to preprocess and prepare it for further use, and to persist it into a local datastore. Preprocessing in this context means applying various language processing techniques on the data. Our general approach and the techniques are presented in detail in section 2. In a second step we show how this data can be presented to the user such that it can easily be accessed and used by humans. As a means of data representation several options are given. The closest and most flexible access path is an API interface to directly query for certain data. In particular, our system allows for multi-dimensional queries, which for instance enables searching and linking *time, location and topic*. Alternatively, to provide intuitive access for users that are not experienced in programming languages we have created a simple web interface on top of our API. The highest layer of visualization is given by a Google Maps Flash application especially designed to present a particular query result. We have used the interest of Germany in other countries (deduced from the cabinet papers) to present an intuitive visualization for humans to make use of this data.

Our choice for the protocols of the German federal cabinet was based on several reasons. Most important are availability and comprehensibility of results. Unlike other historical data sources the meeting protocols are available online as RTF formatted files. Results extracted from the corpus can easily be verified due to the well documented time period after the second world war, especially for Germany.

The final evaluation allows the conclusion that existing language processing tools - applied correctly - can well be used to significantly facilitate the procedure of data extraction from historical corpora that are due to their size unmanageable without further tools.

This paper is organized as follows. First we introduce necessary background technologies used in the project implementation and propose our general approach. Then we present our implementation in detail and give an evaluation on our results. Finally, we draw a conclusion and present some open issues designated for future work.

2 BACKGROUND & APPROACH

In our system we make use of various common language processing and text mining technologies. In particular we develop a general approach how to integrate them into a data mining system capable of processing large data sources by computational means. In the following sections we first present several important technologies and their implementations, and then introduce and explain our approach.

2.1 Technologies

Various language and data mining techniques have been developed and published. For most of them open source implementations and solutions are available. Some tools that can be used for our approach are introduced here.

Tokenization Tokenization describes the process of splitting some input text into meaningful parts called tokens. Besides naive approaches such as whitespace tokenization there are more advanced

[1] Universität des Saarlandes, Saarbrücken, email: firstname.lastname@stud.uni-saarland.de

[2] http://www.bundesarchiv.de/cocoon/barch/0000/index.html

methods available [3]. The OpenNLP tokenizer[3] is one particular example: trained on some language model it is able to return an intuitive token stream. In our implementation we use this tool at index construction time.

Stemming When computationally processing text it is often desireable to find the root of a word in order to determine whether different words belong to the same root or not. Various algorithms exist which implement a stemming functionality.

In this project we make use of Snowball, "a small string processing language designed for creating stemming algorithms"[4]. It was implemented by Dr. Martin Porter, one of the pioneers of stemming [5]. The implementation needs to be fine-tuned to each individual language, and applies a fixed procedure of strict rules to find the stem of a word.

Topic classification Texts usually talk about a certain topic. Quite frequently even single texts contain several topics. The idea of classification is to assign one or more topics to some textual input. However, this is not an easy task at all, since it often proves to be difficult even for humans. Given a large sample set of documents human classification is often inconsistent even if categories are given.

One particular tool that can be applied for this purpose is the LingPipe suite[5]. LingPipe is a Java framework that provides various tools for processing human language. Besides topic classification it provides part-of-speech tagging, entity tracking, spelling correction, and many more features. In our project we use the LingPipe API as a classifier. [4], [7] and [8] discuss technical insights to this.

2.2 Our approach

We propose a general five step approach to the problem of tackling large data sources by computational means:

1. **Retrieval of raw data:** First the data needs to be retrieved from the data source to the local hard disk. The commonly used technique, which is also applied in our project, is focused crawling.
2. **Preparing access path to raw data:** Commonly the data is available locally in an unstructured way, like for instance in plain text format. The idea of this step is to provide a high level access path to the raw data.
3. **Preprocessing:** The most important phase in this approach is preprocessing the raw data. Various language processing techniques are applied to process the data and prepare it for further use. Our proposed approach suggests building various inverted indices for efficient search first [9]. In order to build these indices we use tokenization techniques, as well as language specific stemming and pruning of stop words. Second, we make use of a trained classifier to categorize the input documents into a set of user defined classes. Afterwards the links between related data instances need to elaborated, and finally the preprocessed data needs to be persisted to be used by the backend. It makes sense to use persistence as the preprocessing process is computationally expensive and actually needs to be done only once.
4. **Implementing the backend:** Having preprocessed the data it can be used and accessed. The backend implementation provides means for this and implements a query and filtering engine. For querying the constructed inverted indices can be used, and for the filtering we employ set operations on the fetched input results. Obviously in this step optimization modules like for instance a query optimizer can tremendously influence performance.
5. **Interfaces and visualization:** The final step in our approach is to build interfaces and presentation layers. We distinguish between three classes of interfaces: Most basic is an API interface to directly communicate with the backend. The next higher interface level is an intuitive web interface, and third special visualization applications for particular purposes. The chosen interfaces obviously depend on the intended use and are application specific.

[3] http://opennlp.sourceforge.net/
[4] http://snowball.tartarus.org/index.php
[5] http://alias-i.com/lingpipe/index.html

2.3 Related work

Several similar projects exist, which focus on retrieving data from historical corpora and the visualization of findings. With regard to our GoogleMaps visualisation two projects are worth mentioning. The GeoCrossWalk project [1] processes the Stormont papers to extract the location the papers are talking about, and produces a map visualization of the results. The GeoDigRef project [6] enriches historical archives with georeferencing information.

In comparison to these projects the main focus of our implementation is at a lower level. The main idea of the project presented in this paper is to provide the option to query for information in multiple dimensions. Then, in a second step a visualization of the resulting data can be generated. Our prototype system for instance allows queries like 'who attended a certain topic', or 'which countries are mentioned how often in year xx'. Our GoogleMaps interface is only one particular visualization of the data that can be obtained by the query interface.

3 IMPLEMENTATION

The remainder of this paper deals with our implementation of a text mining system. For this prototype solution we decided to use the federal German cabinet meeting protocols from 1949 to 1964. However, the presented techniques can easily be applied to any other dataset. In the following we discuss how we applied our proposed approach to build such a system. In particular we demonstrate that a system like ours can be built with quite little effort using various available tools.

Retrieving and preparing the data As a first step the data must be received and prepared for further access. If no public interface to the dataset is available, the natural approach is crawling and parsing. This is also our approach of choice for the given project. For the major part of the crawling we used the lobobrowser toolkit[6] which allows access to the DOM (Document Object Model) tree of arbitrary web sites. Fortunately the data in the federal German archive is available in a tree-like structure. Hence, we were able to collect the data and simultaneously collect valuable information such as for instance the date of the meeting. In the federal German archive the protocols of each agenda item were provided in RTF documents. The process of data retrieval took about seven hours for this dataset, however, we assume that most of the time was spent waiting for the server to reply.

Having collected the raw data we prepared it for further access. In our Java based system we provided a simple interface that allows for accessing relevant plain text information as well as meta information.

[6] http://lobobrowser.org/java-browser.jsp

Preprocessing Since the retrieved data is available in a rather unstructured way, further data processing is necessary. During this preprocessing phase our system generates inverted indices, performs topic classification and moreover provides links between related data instances. Some details about this are presented in the following.

Our intention of the system is to provide a flexible and intuitive query interface to the data. One standard technique to allow for efficient query computation on search terms is an inverted index. For the construction of our index we applied the OpenNLP tokenizer using their German token model to generate a stream of tokens. We want to point out that naive approaches such as whitespace tokenization result in very poor results (e.g. "my house." becomes {my, house.}) and emphasize the necessity to apply more sophisticated methods like the OpenNLP tool. Each found occurrence mapping of these tokens is then entered to our index. Note that we used a list of common German stopwords to reduce the term space and to improve search results. A second language processing technique we applied here is stemming: By means of the Snowball stemming algorithm for the German language we enable the system to recognize related search terms which refer to the same concept. Hence for instance the search terms *Katze* (cat) and *Katzen* (cats) will both match to the same concept, as the stemmed transformation of *Katz* is used in index construction and query evaluation.

The second major preprocessing step is that of topic classification. We use the classification module of the LingPipe Framework for this task. To train this tool we collected a number of documents from the corpus for each of our categories[7]: many of the agenda item protocols provide information about the ministry which was responsible for the given item and we used this information to infer the topic of that protocol semi-automatically. This inferred training material was then used to form a separate training data set which we applied as input during initialization of our classifier. Two examples for a category in our application scenario would be those of *Wirtschaft* (Economy) and *Außenpolitik* (Foreign Affairs). Using the trained classifier to categorize each document based on a ranking scheme we obtained the top-2 results. Note that the API internally uses a language model based on n-grams on a word-level (in our case we configured it to use 6-grams). It is worth mentioning that we tested several other classification approaches like a frequency analysis method amongst which the previously described one produced the best results. An example classification for some items is depicted in Figure 1. You can observe that the classification is not perfect, but that it produces quite promising and for humans intuitive results.

The final step of the preprocessing phase is to establish connections between related data instances in our model, e.g. link agenda items to their cabinet meeting instance. In the end the preprocessed information is serialized and stored onto a hard disk for further use by the backend.

The backend For running our backend we use a J2EE compliant application server. The advantage is that we can register our API to the RMI (Remote Method Invocation) service and use web applications at the same time. Since we are using a main memory data model we have to run the server instance with a sufficiently high maximum heap memory setting. We describe the API interface to our system in the following paragraph.

Our backend implements a powerful query- and filtering engine

[7] For our dataset we inferred the following set of categories from the occurence of ministries: Außenpolitik, Gesundheit, Innenpolitik, Innenpolitik-Staat, Innenpolitik-Volk, Justiz, Landwirtschaft, Verkehr, Verteidigung, and Wirtschaft

Handelsabkommen mit Uruguay --- [Außenpolitik, Wirtschaft]
~ Trade agreement with Uruguay --- [Foreign Affairs, Economy]
Bericht über die Verhandlungen in Paris --- [Verteidigung, Außenpolitik]
~ Report on negotiations in Paris --- [Defense, Foreign Affairs]
Tarifmaßnahmen im Omnibuslinienverkehr --- [Verkehr, Innenpolitik]
~ Measures on rates in public transport --- [Transport, Home Affairs]
Durchführung des Teesteuer-Gesetzes --- [Justiz, Wirtschaft]
~ Laws concerning taxation of tea --- [Justice, Economy]
Entwurf einer Verordnung über Zolländerungen --- [Verkehr, Gesundheit]
~ Draft of new directives in customs --- [Transport, Health]

Figure 1. Classification of 5 agenda items

which can be used to invoke arbitrary queries on the data. Queries can be expressed in conjunctive forms of interconnected query and filter attributes. Suppose for instance a query asking for all agenda items which were discussed during the year 1950, are classified as *Wirschaft* (Economy), and talk about a certain country. This query could be invoked as illustrated in Figure 2.

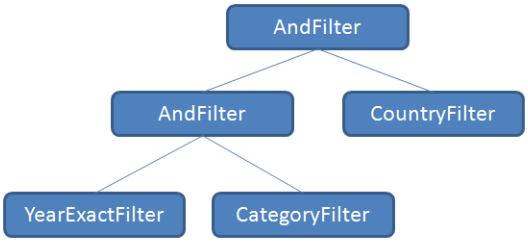


Figure 2. Example for a three item Filtering

Even though our backend is very powerful, it has some limitations. This is mainly due to its prototype character and design as a proof-of-concept. As mentioned already our system needs quite a large amount of main memory. This could be reduced by using more efficient data structures and a different data model. The second remark regards scalability and performance: in the current implementation we use simple Java collection classes for the whole data management and all data operations (e.g. set operations, etc.). There are obviously more sophisticated solutions that can be used to increase the backend performance.

Interfaces to the system We provide three interfaces to our system: A Java interface that can be accessed via RMI, an intuitive and flexible web interface, and third a Google Maps visualization of the Year vs. Country relationship. The Java interface provides the most powerful access to our system as the Java programming language can be used to directly work with the data. An example scenario could be the *Year / Country Analysis* which computes country occurrences grouped by year (Figure 3). Note that the results of this query are used for the Google Maps Flash application discussed below.

The web interface provides a graphical user interface layer on top of our system. It provides an intuitive query interface, a browsing feature and also presents various findings. By means of a simple form

```
TextMiningApi api = (TextMiningApi) Naming.lookup("rmi://localhost
    :60501/backend");

System.out.println("List of countries occurring in the data:");
List<String> countries = api.getCountries();
for (String s : countries)
    System.out.println(s);

System.out.println("Occurrences per year:");
for (int i=1949; i<=1964; i++) {
    String year = Integer.toString(i);
    System.out.println("### YEAR " + year + " ###");
    YearExactFilter yearFilter = new YearExactFilter(year);

    for (String s : countries) {
        List<Document> res = api.getDocuments( new AndFilter(
            yearFilter, new CountryFilter(s)));
        if (res.size() > 0)
            System.out.println(s + ": " + res.size());
    }
}
```

Figure 3. Source Code: Year / Country Analysis

the user can perform any kind of multi-dimensional AND query. An example sequence of screenshots for the query "Familie" and "1954" is given in Figure 4. Moreover a browsing facility for explorative access is provided.

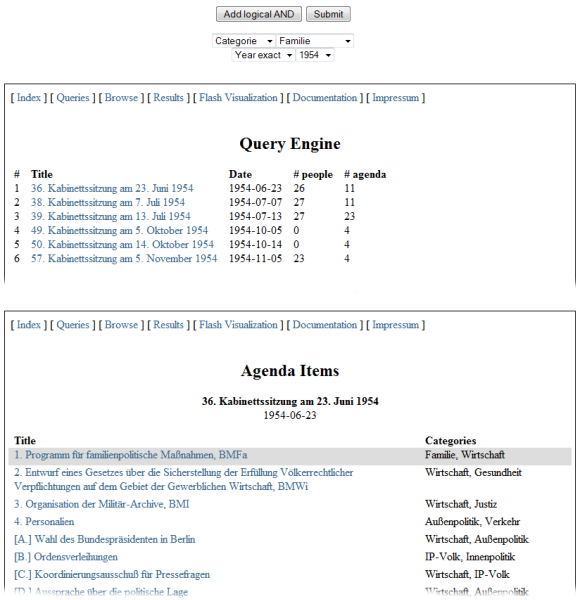


Figure 4. Example usage of the Query Interface

Third, for visualization purposes of a particular use-case we employed the Google Maps API and Flash [2] to illustrate the Year vs. Country analysis. This is illustrated in Figure 5. A timeline allows the user to navigate between the years from 1949 till 1964. For each chosen year the world map is colored in accordance to the data values returned by the query mentioned above. The more often a country was mentioned in the cabinet meetings, the more intense is the color which overlays the map. The user can toggle between a world view and a close up of Europe to distinguish the different European countries more easily.

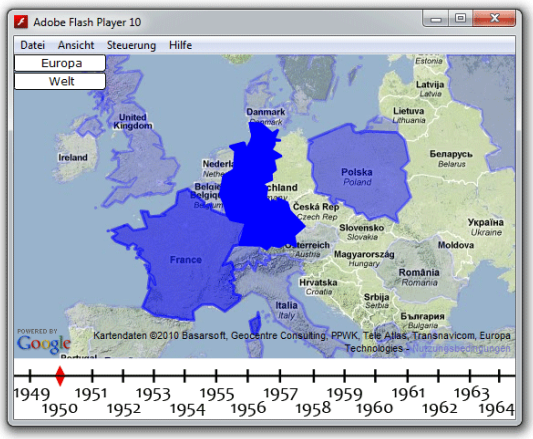


Figure 5. Google Maps Visualization - European View

4 EVALUATION

The project results can be evaluated as follows. Using the topic classification we can find hot spots and trends in the agenda items. Figure 6 shows one interesting inverse correlation: At times when foreign affairs required increasing attention by the German ministers, it seems they spent less time discussing home affairs. In Figure 6 the red (upper) graph depicts the number of times a topic related to foreign affairs was discussed, the green (lower) graph shows the same for home affairs.

We checked the produced results against some well known historical events. Our algorithm shows an increase in foreign politics and defense related meetings around the time of construction of the Berlin Wall and the Cuba crisis. Both increases can be expected given the influence those events had on Germany.

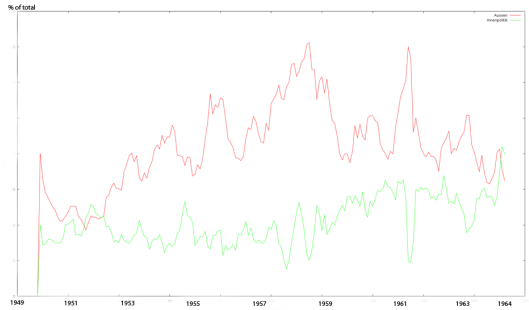


Figure 6. Correlation of foreign (red, above) and home affairs (green, below)

In a second evaluation we compared the automated topic classification to manual classification. To this end we classified a small sample subset of 100 items manually and compared it to the automated findings. We found a high correlation between our classification and the ones provided by the prototype project implementation. Out of the 100 manually classified agenda items, 87 could be used for further evaluation. The other thirteen items were very short statements - like announcements of staff decisions - where classification based on title information only is too difficult. The remaining 87

data items can be evaluated for the number of correct classifications. For 38 agenda items both classifications were correct, for additional 40 agenda items the first of two classifications was correct. Only 9 agenda items were classified incorrectly.

In summary almost 90 percent of the items were classified correctly if only the first classification is used. Most of the wrong second classifications were due to the fact that many items can be clearly related to a single topic only: so the second classification is by default not more than a good guess by the machine.

As a specific use case we have designed a visualization for the query of "country occurrences during certain years". With the data returned from this query a map is colored according to the interest of Germany in a given area at a certain time (see Figure 5). The intensity of the blue color is proportional to the number of references in the German cabinet meetings. We verified the plausibility for a few countries. Most visible is the interest of Germany in the United States, which can easily be related to the post WWII role of the US for the reconstruction of Germany. The relation to France and Britain - which were occupying powers and close political partners after 1954 - can also be seen in a dark blue color throughout the given time span.

The data from the attendance lists together with the classification of agenda items allows queries like 'Who attended meetings about defense related topics in November 1951'. This data was verified for a couple of samples and found to be complete and correct as well.

5 CONCLUSION & FUTURE WORK

Based on the evaluation we can conclude that our prototype implementation presents a functional and powerful tool for processing large amounts of data. However, due to time constraints we had to limit our requirements and feature set to a certain degree. The following listing provides information about some open issues and other ideas:

- **Performance:** as we implemented a prototype system, we provided a proof-of-concept solution. There are many optimizations possible, one is for instance to use more sophisticated data structures and merging algorithms within the query and filtering engine.
- **Scalability:** our solution is only scalable to a certain degree. This is due to the implementation of our system following a main-memory data model. Moreover, the issues mentioned for performance are a factor as well.
- **Interfaces:** the Java interface can be extended with further features and more convenience functions. Furthermore, the web UI is a very simple approach to visualize our findings. For a real world application it would be absolutely necessary to make it more robust, and to add further features for more flexibility.
- **Analysis:** some further analysis might be necessary for correctness of all countries, persons and meetings. This regards for instance encoding problems and certain spelling mistakes.
- **Named entity recognition:** in a future version named-entity recognition should be implemented to distinguish persons, countries and items necessary for the topic classification.

ACKNOWLEDGEMENTS

We would like to thank Dr. Caroline Sporleder for her support and valuable input. Our gratitude also goes to the reviewers for their time and constructive criticism of the paper. Furthermore, we are very much obliged to the Saarbrücken Graduate School of Computer Science for their financial and ideological support concerning the conference contribution and attendance.

REFERENCES

[1] S. Anderson and S. Dunn. Embedding GeoCrossWalk, October 2009. http://mykcl.info/iss/cerch/projects/portfolio/embedding.html.

[2] Google. Google Maps API for Flash , May 2010. http://code.google.com/intl/en/apis/maps/documentation/flash/intro.html.

[3] P. Jackson and I. Moulinier, *Natural language processing for online applications: Text retrieval, extraction and categorization*, John Benjamins Pub Co, 2007.

[4] F. Peng, D. Schuurmans, and S. Wang, 'Language and task independent text categorization with simple language models', in *NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pp. 110–117, Morristown, NJ, USA, (2003). Association for Computational Linguistics.

[5] M. F. Porter, 'An algorithm for suffix stripping', *Program*, **14**(3), 130–137, (1980).

[6] J. Reid (Project Manager). GeoDigRef, August 2009. http://edina.ac.uk/projects/GeoDigRef_summary.html.

[7] F. Sebastiani, 'Machine learning in automated text categorization', *ACM Comput. Surv.*, **34**(1), 1–47, (2002).

[8] W. J. Teahan, 'Text classification and segmentation using minimum cross-entropy', in *Proceeding of RIAO-00, 6th International Conference "Recherche d'Information Assistee par Ordinateur"*, Paris, FR, (2000).

[9] J. Zobel and A. Moffat, 'Inverted files for text search engines', *ACM Comput. Surv.*, **38**(2), 6, (2006).

Similarity-Based Navigation in Visualized Collections of Historical Documents

Yevgeni Berzak[1] and **Michal Richter**[2] and **Carsten Ehrler**[3]

Abstract. Working with large and unstructured collections of historical documents is a challenging task for historians. We present an approach for visualizing such collections in the form of graphs in which similar documents are connected by edges. The strength of the similarities is measured according to the overlap of historically significant information such as Named Entities, or the overlap of general vocabulary. The visualization approach provides structure that helps unveiling otherwise hidden information and relations between documents.

We implement the idea of similarity graphs within an Information Retrieval system supported by an interactive Graphical User Interface. The system allows querying the database, visualizing the results and browsing the collection graphically in an effective and informative way.

1 INTRODUCTION

The availability of historical documents in digital form has been constantly increasing in recent years. Digitization of sources is extremely valuable for historians, as it contributes to preservation, facilitates accessibility and enables exploiting computational methods. Despite the growth in volume of digitized historical data, available collections are rarely accompanied by supporting tools which significantly facilitate the work of historians.

The existing interfaces typically include a simple keyword or metadata search, providing users with a list of documents that match their query. Though such tools can indeed spare valuable time dedicated to manual work, they are far from exploiting the full power and flexibility that state of the art Natural Language Processing (NLP) and Information Retrieval (IR) technology has to offer. Moreover, the systems provided are usually generic, and rarely address the specific needs of researchers in the historical domain. This situation calls for the development and adaptation of NLP techniques for historical data, as well as for the creation of user interfaces that would enable historians to use this technology effectively, in a way that would meet their needs.

This work addresses both aspects of the current shortage. We take up the NLP domain adaptation challenge by applying and tailoring NLP tools that extract information relevant for historians and create links between documents according to similarity with regard to this information. The need for intuitive user interfaces is addressed by providing an interactive graphical tool that enables historians without computational background to benefit from NLP technology. This twofold approach aims at improving the chances of historians working with digitized sources to find information that is relevant for their research goals.

The core idea of our approach is to extract information with special importance for the historical domain, such as Named Entities (NEs) of the types *persons*, *locations* and *organizations* mentioned in each document, and then use this information to determine similarity between documents. Our working hypothesis is that the higher the similarity between two documents according to NEs, the more probable it is that they are related to each other in an interesting way. In addition, we also use a generic, domain independent similarity measure based on the remaining vocabulary of the collection. These different similarity measures can be used separately or in combination with one another. The measured similarity rates can be interpreted as strength of potential connections between documents and visualized as edges of a graph.

We incorporate this idea in an IR system wrapped by an interactive Graphical User Interface (GUI) that provides powerful search operations and allows for visual navigation through collections of historical documents. We exemplify our approach on a collection of speeches and other oratory materials by Fidel Castro. Our system includes keyword and metadata search. Retrieved documents are presented in a table synchronized with an interactive scalable graphical network that connects related documents. This representation is designed to support effective visual navigation in the collection, allowing identification of documents which revolve around similar topics, distinguishing relevant from irrelevant documents, and also enabling exploration of specific relations between any subset of the retrieved documents. Our system can also be viewed as a recommender tool: given an interest in a specific document, the user can easily identify documents that are most similar and hence potentially related to it. To the best of our knowledge this is the first system of its kind.

This paper is structured as follows. Section 2 presents related work and background. Section 3 describes the dataset we use and the information extraction process. In section 4 we elaborate on the similarity measurements and the visualization of the collection according to these measurements. Section 5 describes the GUI that realizes our visualization approach and in section 6 we illustrate its use for historians and exemplify its advantages. Finally, we discuss future research perspectives in section 7.

2 BACKGROUND

This work can be located within the field of Domain Knowledge Visualization. This field is centered around visualization techniques of

[1] Department of Computational Linguistics, Saarland University, Germany, email: jeniabk@gmail.com

[2] Institute of Formal and Applied Linguistics, Faculty of Mathematics and Physics, Charles University in Prague, Czech Republic, email: michalisek@gmail.com

[3] Department of Computer Science, Saarland University, Germany, email: carsten.ehrler@googlemail.com

domain structures, in particular of scientific domains. Its notable applications are mapping structures of domains and supporting IR and information classification. The general process flow of visualizing domain knowledge as described in [3] is the following:

1. Collection of data (i.e. extraction of documents belonging to the domain)
2. Definition of unit of analysis (e.g. documents, authors, terms)
3. Selection of similarity measures and calculation of similarity between units
4. Ordination or assignment of coordinates to units
5. Use of resulting visualization for analysis and interpretation

Our work-flow conforms to these categories. We work with a historical database and use documents as the units of analysis. We measure similarity between documents according to extracted terms overlap with the cosine metric. Finally, we visualize subsets of documents, selected according to a specified query, as graphs ordinated with a force based layout algorithm.

We use tools and algorithms for NE Recognition and string similarity in order to extract the information according to which the inter-document similarity is measured. The Vector Space Model (VSM) serves as our framework for representation of documents in the collection. The VSM enables straightforward modeling of inter-document similarity as proximity of vectors in the multidimensional vector space. Furthermore, this representation is a standard approach in IR, allowing ranked retrieval of parts of the collection that match the user's query.

Our graphical model responsible for ordination of the documents is reminiscent of Pathfinder Networks (PN) [7]. Modeled as PNs, collections of documents may be presented as graphs that capture the relative proximities of objects to each other. Objects are represented as nodes and proximities between objects are represented as links between nodes, where only the strongest links according to an adjustable proximity threshold are taken into consideration. Proximity can have several interpretations, one of them being similarity. In our model, degree of similarity is reflected in the thickness of the edges. It also has impact on the strength of the attracting force between linked nodes in a Force Directed Placement algorithm using which the graph layout is determined. This approach provides an aesthetic, intuitive and transparent representation that conforms to the similarity relations between the documents.

3 INFORMATION EXTRACTION FROM A HISTORICAL COLLECTION

Our approach to detection of historically useful relations between documents focuses on exploiting domain relevant information for similarity measurements. An important type of such information in the historical domain are NEs. In this work we address three types of named entities: *persons*, *locations* and *organizations*. All three play an important role in historical documents in general, and collections of modern political history in particular. Moreover, these are well studied classes of NEs [6] and existing tools for their identification and classification perform well. In the following we describe our data and elaborate the information extraction procedure.

3.1 Dataset

Our case study for a collection of historical documents is the Castro Speech Database (CSDB) [1], maintained by the Latin American Network Information Center (LANIC) at the University of Texas at Austin. This collection contains 1492 English translations of speeches, interviews, press conferences and other oratory materials by Fidel Castro from 1959 to 1996. Most of the documents in the database are annotated with metadata, including document type (e.g. *speech*, *interview*, *meeting*), date, place, author, source, and headline.

The documents are manually translated from modern Spanish into modern English, sparing many problems associated with documents written in historical languages. This characteristic, along with a considerable amount of newswire content allow a relatively straightforward application of tools and models already deployed and currently used in NLP and IR.

However, the collection is very heterogeneous and comprises different genres and styles. Additionally, many documents in the corpus are based on spoken language, featuring heavily rhetorical content and vague structure. Given these characteristics, identification of useful relations between documents is a particularly challenging task.

3.2 Extraction of named entities

In order to recognize NEs in the documents of the CSDB we apply the Stanford Named Entity Recognizer [4]. This tool is based on Conditional Random Fields (CRF) using a wide range of features. It is available with pre-trained models and it is robust across domains, a property that is highly desirable for our diverse database.

For the purpose of computing the similarities, the recognized NEs can be regarded as reduced forms of the original documents. Following the extraction, the documents of the collection $D = \{ d_1, d_2 ... d_{\text{N}} \}$ are indexed as vectors in four distinct term spaces $\mathcal{T} = \{ T_{\text{PER}}, T_{\text{LOC}}, T_{\text{ORG}}, T_{\text{VOC}} \}$ corresponding to the three types of named entities extracted from the collection and an additional term space for the general vocabulary. The general vocabulary term space contains all the content words in the collection that do not belong to the NE terms spaces. We consider two standard weighting schemes for measuring the importance of a term $t \in T$ for a document $d \in D$, namely TF and TF/IDF. An index matrix $I_{\text{T}} = \mathbb{R}^{|D| \times |T|}$ is computed for each term space $T \in \mathcal{T}$ and a weighting scheme $w \in \{\text{TF}, \text{TF/IDF}\}$. Each position in the matrix $I_{i,j}$ contains the score w of term $t_j \in T$ in document $d_i \in D$.

3.3 Aliasing

Relying on the raw output of a NE recognizer might not be sufficient if we desire reliable counts of NEs. One of the major problems is that the same NE can be manifested in a variety of phrases. In order to know how many times a certain NE appears in a document, we should be able to identify all the phrases that refer to this entity. This task is known as Coreference Resolution. Coreference Resolution is a complex problem and state of the art tools achieve only moderate accuracy on this task. Due to the limitations of existing technology and the nature of our task we restrict ourselves to multi-document aliasing, that is, recognizing linguistic variants of names in a collection of documents.

Our approach to aliasing relies on string similarity. We measure similarity between NEs that belong to the same class using a String Kernel (SK) [8]. SKs are kernel methods which operate on the domain of strings over a finite alphabet Σ and are commonly used in the Bioinformatics domain.

In this work, a p-spectrum SK is used. A p-spectrum SK is defined in equation (1), where $\phi_u^p(x)$ counts the number of occurrences of the

substring u of length p in string x.

$$sk(s,t) = \sum_{u \in \Sigma^p} \phi_u^p(s)\phi_u^p(t) \quad (1)$$

The associated Hilbert Space for this kernel is $\mathbb{R}^{|\Sigma^p|}$, the space of all possible strings over Σ of length p. The p-spectrum SK can be efficiently implemented by using trie data-structures over n-gram models.

The advantage of this method over more traditional string similarity metrics such as Edit Distance is its flexibility, especially with regard to word order. For example the strings *Public Health Ministry* and *Ministry of Public Health* which have a big Edit Distance, are highly similar according to the SK. Unfortunately, this flexibility often leads to over-generation, by also considering strings such as *Polish People's Republic* and *Lao People's Republic* as highly similar. The over-generation can be lowered to some extent by setting a high similarity threshold for considering two NEs as aliases of each other.

Using the p-spectrum SK a similarity matrix $K \in \mathbb{R}^{|T|\times|T|}$ is computed for each class of NEs $T \in \mathcal{T}$, where $K_{i,j}$ is the the SK similarity measurement of terms t_i and t_j. Similarity values below a predefined threshold are discarded.

The aliased document vectors are computed by multiplying the similarity matrix with the original document vectors. In this way, aliases of names that appear in the documents receive an additional weight. A similar expansion is used for queries that contain NEs. An example for the aliasing method is provided in figure 1. The importance of a NE in the aliased documents is influenced by the number of its aliases. To prevent this effect, it is possible to perform normalization of the similarity matrix or the aliased documents.

$$\underbrace{\begin{pmatrix} 1 & 0.8 & 0 \\ 0.8 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{K} \underbrace{\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}}_{d} = \underbrace{\begin{pmatrix} 1 \\ 0.8 \\ 0 \end{pmatrix}}_{\bar{d}}$$

Figure 1. Aliasing of a document with a SK similarity measure. The matrix K is the kernel similarity matrix of the three terms $T = \{$*Fidel Castro*, *Dr. Fidel Castro*, *Raul*$\}$. A document vector $d = (1, 0, 0)^{\mathrm{T}}$ that contains only the term *Fidel Castro* is expanded into an aliased form $\bar{d} = Kd$, in which the alias *Dr. Fidel Castro* has a non-zero weight

.

The aliased document representations serves two purposes. First, we receive more reliable similarity measures between documents. Secondly, the flexibility of the querying mechanism is increased by expanding NE terms in the query to all their aliases, allowing retrieval of name variations for query keywords that are NEs.

4 VISUALIZATION OF DOCUMENT SIMILARITIES

Given the vector representations of documents, we can obtain their pairwise similarities, and present them graphically according to this information. However, it is impossible to visualize multidimensional vectors directly, thus such representation must be reduced to two or three dimensions, a process often referred to as *ordination* [3]. In this section we elaborate on our ordination approach for the presentation of multidimensional vectors in a 2D space that adheres to their similarities.

4.1 Similarity measurement

Using the constructed indexes described in section 3.2, we measure and store the similarity for each pair of documents in the collection. To determine the similarity we use the standard cosine measure as defined in (2).

$$cosim(v, v') = \sum_{t \in T} \frac{v(t)v'(t)}{|v||v'|} \quad (2)$$

It expresses the cosine of the angle between the document vectors v and v'.

Separate similarity matrices are computed for each combination of term space, indexing scheme and aliasing setting. The similarity matrices of the different term spaces can be combined. For this purpose a weight vector $w = [w_{\text{PER}}, w_{\text{LOC}}, w_{\text{ORG}}, w_{\text{VOC}}]$ is defined. The final similarity matrix is computed as a linear combination of the similarity matrices of the different term spaces according to the specified weight vector. By setting $w = [0.5, 0.25, 0.25, 0]$, we express that *persons* are twice as important as *organizations* and *locations* and that the general vocabulary is ignored.

4.2 Visualization of similarities

4.2.1 Force directed placement

To visualize the documents in a 2D graph we represent documents as nodes and encode the similarity rates between pairs of documents as weights for edges that connect them. An edge is established only if the similarity between a pair of nodes exceeds a certain threshold. Having this specification we apply a Force Directed Placement (FDP) algorithm [5], which arranges the otherwise unordered nodes according to the similarity relations between them.

FDP algorithms are a class of algorithms that is based on physical modeling of elements. In particular, nodes are modeled as physical objects that are electrically charged with the same sign, and the connections between them are modeled as springs. Therefore, the nodes repel each other causing the graph to spread, while the springs try to group related nodes together according to the spring force and an optimal spring length. In our implementation, the magnitude of the spring force is defined with regard to the measured similarity rate between the nodes it connects. In each iteration of the algorithm, the forces that act on the nodes cause them to rearrange accordingly. After several iterations, the position of the nodes becomes (quasi) stationary and the system stabilizes in a local equilibrium state.

In the resulting graphs similar nodes tend to be closer to each other then dissimilar ones. To provide further transparency of the similarity rates, the thickness of each edge corresponds to the similarity measurement between the pair of documents it connects: the stronger the similarity, the thicker the edge. This gives an indication for the strength of each edge regardless of its length.

We believe that this approach provides a clear, intuitive and aesthetic visual representation of the documents. Moreover, it is flexible with regard to the positioning of the nodes.

4.2.2 Graph clustering

One of the major advantages of our visualization approach is the ability to identify groups of documents that are highly similar to each

other. Such groups may be informative for inferring common topics or help filtering out documents less relevant for the interests of the user.

Given the nature of our graph representation it is often possible to identify groups of strongly interconnected documents without any manipulation of the graph. However, in many graphs an automatic identification of dense regions and re-arranging of the graph according to these regions might be useful.

In order to enable this functionality, the Chinese Whispers clustering (CWC) [2] is used. CWC is a non-parametric algorithm that is applied on the nodes of weighted undirected graphs, such as our data structure. Since we do not know in advance the number of clusters that will emerge from our data, the property that CWC is non-parametric is desirable. This algorithm is also very efficient: its time complexity is linear in the number of edges.

The CWC algorithm works in a bottom-up fashion. During initialization, each node is assigned with its own cluster. The algorithm then performs a number of iterations in which each node is sequentially assigned to the class that has the highest sum of weighted edges to the node. After few iterations a mostly stable clustering is achieved with at most few nodes for which the algorithm might continue changing the class assignment.

The output of CWC can be further used as an initial setup for the FDP algorithm. Initialization of node positions in such a way that nodes belonging to the same cluster are close to each other enables FDP to converge to a better layout.

Clustering with CWC is a powerful and efficient way of enhancing the informativeness of our visualization approach. In the following section we demonstrate how this approach can be integrated in a GUI that enables users to utilize it for their needs.

5 GRAPHICAL USER INTERFACE

We incorporate our approach for visualizing collections of historical documents in a GUI. The GUI allows the user to query for keywords and present the outcome of the query as an interactive graph, based on the description in section 4. The GUI implements additional features that enable customization of the graph and aim at maximizing the flexibility and benefit that historians can gain from using our approach. We consulted a historian to ensure that our design indeed addresses the practical needs of end-users. Figure 2 shows a screen shot of the entire system. We present the main characteristics and features of the GUI in this section.

Our system provides an IR mechanism, where the user can specify both query terms (feature 1 in figure 2) and constraints on metadata, e.g. dates and type of document (feature 2). Query terms are transformed to a vector model representation of the query. Metadata constraints are translated to database queries. Documents that do not fulfill the metadata constraints are filtered out. The relevant documents are then sorted according to their cosine similarity to the query vector. The specified number of most similar documents is retrieved as a search result. The queries can be expanded using the aliasing method described in section 3.3 to contain aliases of the NE query terms.

The top results of the query are displayed both in a table (top panel) and as a graph (central panel). The table, which is perhaps the more traditional way of receiving search results, lists the results ranked according to their relevance to the query. The table contains all the metadata information of the documents and allows the user to identify them easily. The graph visualizes the connections between the documents in the table. The table and the graph are synchronized, i.e. when a document is chosen in the table it is marked in the graph and vice versa.

The graph is scalable (enables zooming in and out) and draggable. The positions of the nodes can be rearranged by dragging them, allowing a manual adjustment of the layout. We also enable automated layouting through the FDP algorithm (feature 6), as described in section 4.2.

The size of the nodes corresponds to their relevance to the query: the bigger the node, the more relevant it is. The shape of the nodes corresponds to the type of document they represent, (e.g. *star* stands for *speech*, *circle* for *report*, *pentagon* for *message*). Selecting a node presents the NEs it contains in a separate panel (feature 3). Selecting several nodes provides the intersection of the NEs contained in the selected documents.

The representation of the edge thickness is quantized: edges are presented as *dotted*, *normal* or *thick*. A separate menu (feature 4) allows adjustment of parameters related to the edges. For instance, the user can specify the similarity threshold for presenting an edge, as well as thresholds for each edge type. The edge thresholds can be set on a relative scale with respect to the similarities of the presented graph, or on an absolute scale.

Another menu (feature 5) allows filtering of the nodes such that only nodes that are connected to a selected node up to a specified depth are presented. Figure 3 illustrates this feature, where only neighbors and their neighbors (depth 2) of the selected node are presented.

Additional functionalities for each node are available via a context-menu, which has options for viewing the text of the document with color-coded NEs and highlighted query terms, viewing metadata related to the document and launching a new search for the documents most similar to it in the collection. With this search option, the documents are ranked according to their similarities to the focused document and the sizes of the nodes in the resulting graph are proportional to these similarities. The constraints on the metadata can be applied in the same manner as for the normal query term search. To provide an additional indication for the similarities strength to a particular document, the node context menu also allows to color directly connected nodes in a shaded range of colors (e.g. black to white), where darker color indicates stronger similarity (see figure 3).

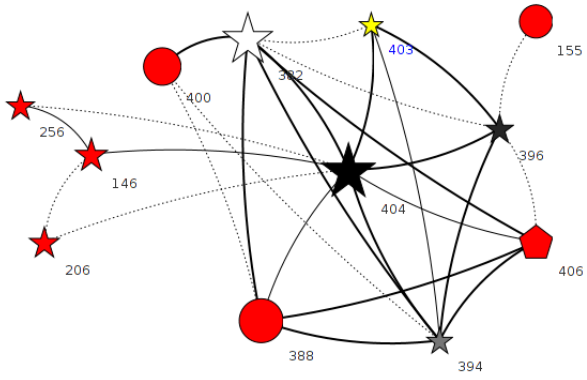


Figure 3. A graph that shows a chosen node (marked in yellow) after coloring the immediate neighbors of the node in a graded color scale according to similarity, and applying depth 2 filtering. The complete graph is presented in figure 2

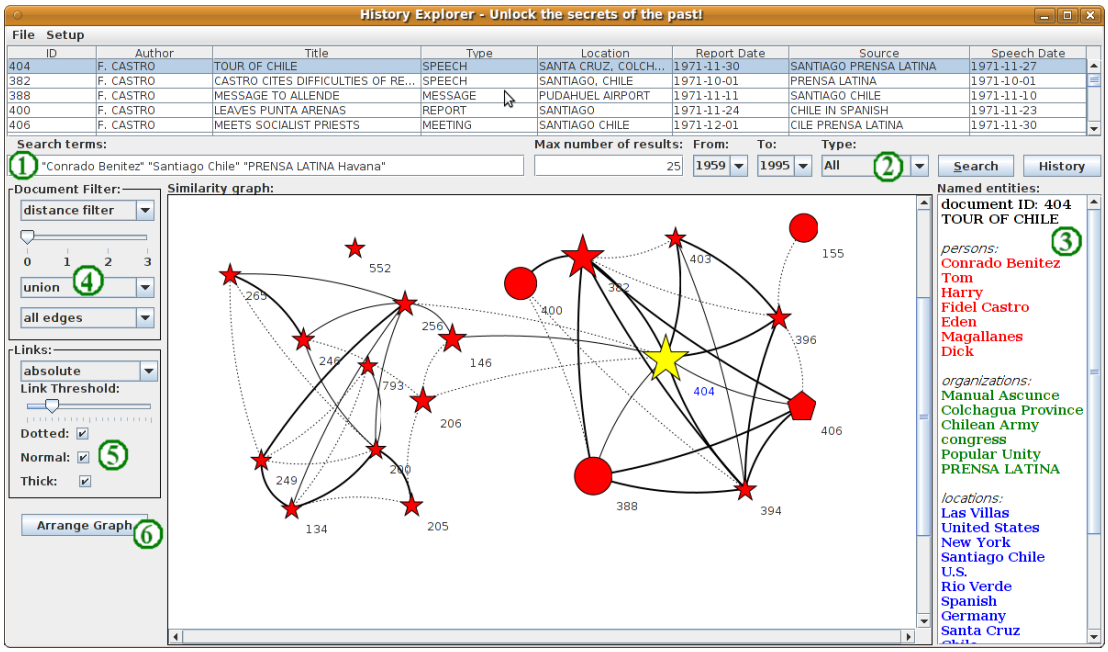


Figure 2. The "History Explorer" GUI

The CWC, described in section 4.2.2, can be applied to a given graph. After the clustering is performed, the graph is redrawn with a layout that separates the clusters and uses a different node color for each cluster. Although the CWC is a non-parametric algorithm, we provide an option to specify the maximum number of presented clusters. If the algorithm constructs more clusters then maximum, only the largest clusters are highlighted in the resulting graph. It is also possible to define a required minimal size of the cluster in order to be highlighted. Figure 4 shows a graph after applying the clustering algorithm.

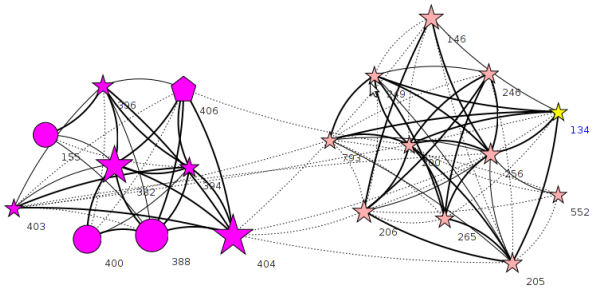


Figure 4. A graph with two distinct clusters that were identified using Chinese Whispers Clustering

The GUI also contains advanced options for setting the weighted combination of the term spaces for the similarity measurements, choice between weighting schemes, enabling and disabling aliasing for search and for similarity measurements, as well as other parameters related to the retrieval and presentation of the graph.

6 THE BENEFIT FOR HISTORICAL RESEARCH

One of the major tasks of historians consists of working with sources that often come in the form of historical documents. Such documents have numerous usages in the various branches of historical research. Some of these are related to the task of revealing general trends and patterns about a historic period or personality while others focus on discovering specific details and pieces of information. Historical documents serve both for the formulation of historical hypotheses and for their validation and rejection efforts.

While the range of written accounts of ancient history is well-known and relatively limited, researchers of modern times often have to face an abundance of written materials. Dealing with large collections of documents introduces additional challenges for historians. In particular, the focus is often shifted to the identification and retrieval of documents that might be of relevance. Furthermore, identifying trends as well as implicit and explicit connections between documents becomes an extremely difficult task if performed manually.

Our system is designed to support historians' work with electronic sources, specifically with large collections of documents such as the CSDB. The system presents the user with a visual *structure* that helps to discover *new knowledge* by highlighting interesting inter-document connections that are otherwise hidden.

The basic idea according to which the graphs are constructed is easily understandable. The produced visualizations allow for an intuitive interpretation that does not require the user to have any computational background. At the same time, they exploit a range of advanced techniques of text processing, IR and Data Visualization that are utilized to produce the desired results.

A particular strength of our system is the ability to combine search and visualization. While the former can be used to express a specific historical question, the latter helps finding answers and formulating

new questions.

Without any undesirable over-simplification regarding the original data, the types of connections or any other information, users receive a representation of the data that can considerably improve their ability to locate, infer and discover relevant information. In this sense, the presented approach is applicable to real research problems in the historical domain. Following are concrete aspects of our work that are likely to be appealing to historians.

We focus on automatic markup of NEs of types that are of potential interest for historians, both in terms of discovery of new entities and identification of known entities.

The IR system includes a query expansion mechanism that allows the retrieval of documents containing form variations of the entities appearing in the query. Our retrieval mechanism and graphical interface incorporates metadata, if such is provided with the documents in the collection. Using metadata further extends the flexibility of the search and allows a more informative graphical presentation of query results.

We present links between retrieved documents based on the overlap of NEs or general lexical overlap. These links can be valuable in many scenarios. For instance, a user interested in a specific document can easily identify which other documents contain similar named entities or similar vocabulary. Linking documents also supports the historian in inferring global statements about the collection or one of its subsets. In particular, one can identify groups of highly interconnected documents. Identified dense regions are likely to reflect different kinds of topics and might be correlated to various parameters, such as events, time and location. Absence of links and identification of "stand-alone" documents can also be highly informative, as they can indicate unique content.

Besides providing structure according to NEs, the application can also help to discover NEs that play an important role in a specified topic or affair that is expressed in the query or emerged as a graph cluster. This is achieved by listing shared NEs in a set of documents. Such information would be very difficult to infer in a big collection of documents that is accompanied only by a keyword search mechanism.

A query example demonstrating some of the advantages of our approach is presented in figure 5, showing the resulting graph for the query "health". The graph includes two groups of interconnected documents. While the bigger group contains documents concerning or strongly related to health-care and health reforms in Cuba in the 80s, the smaller group has three documents that deal with a completely different topic. All three are descriptions of meetings of Castro with officials of the Romanian government. They were retrieved simply because the participants of the meetings exchanges greetings, wishing each other "good health". Grouping and separation between relevant and irrelevant documents is a clear benefit of connecting similar documents that cannot be seen in the standard keyword search. Furthermore, within the group that contains the relevant results, documents that discuss similar specific issues such as hygiene or elderly care tend to be connected more strongly than other documents. Such connections are visible in the thickness of the edges or by marking neighbors of specific documents for graded similarity. Finally, document 583 that is not connected to any other document, indeed describes a somewhat different affair. It is a speech of the Cuban minister of health at a UN conference on the relation between poverty and population growth, accusing the imperialist powers in the demographic explosion in Cuba.

In some graphs that contain large bundles of interconnected documents a further insight can be gained by applying clustering. For

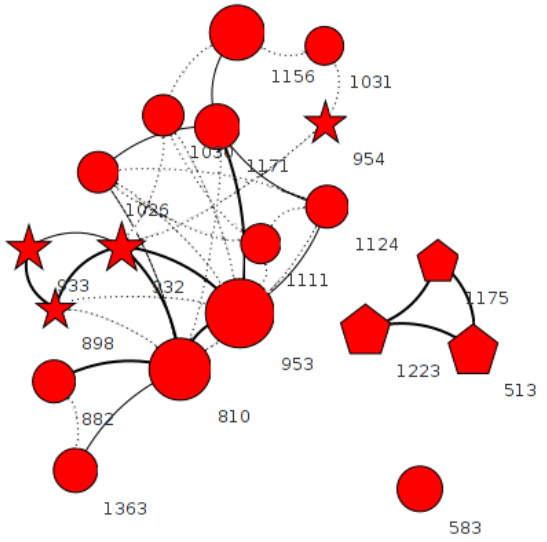


Figure 5. The resulting graph for the query "health", limited to 20 top documents

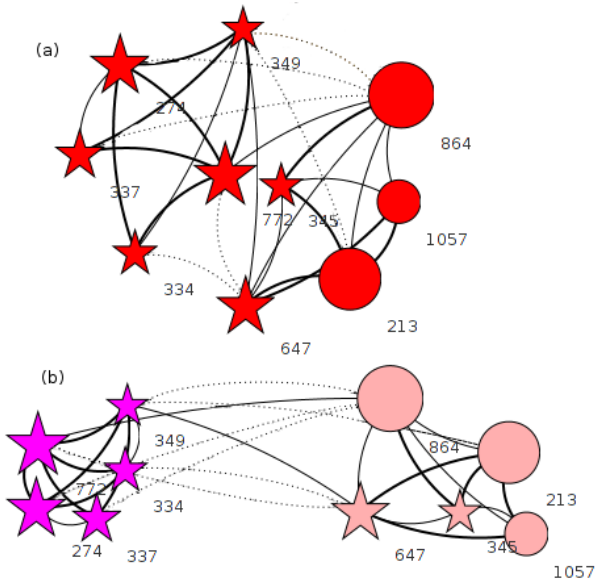


Figure 6. The resulting graphs for the query "Giron Kennedy" before clustering (a), and after clustering (b)

example, the query "Giron" and "Kennedy" produces a highly interconnected graph. This graph can be split into two interpretable clusters using the CWC. Both layouts are shown in figure 6. In the clustered layout, one cluster contains documents directly related to the 1961 Bay of Pigs Invasion, e.g. speeches on the anniversary of the event or victory speeches. The other cluster is composed of documents in which the invasion is mentioned but the document is not directly related to the event. These documents deal with different political and economic aspects of the US-Cuban relations and are thus related to each other. Throughout the different decades during which those documents were composed, the Bay of Pigs Invasion remained a symbol for the hostile nature of these relations.

The provided query examples demonstrate some of the benefits historians can gain from our approach. Nevertheless, it is important to note that in some cases, informative interpretation of the additional information provided by the system is rather challenging.

Overall, our GUI aims to be both intuitive and simple while allowing considerable amount of flexibility. Finally, our system is designed to make retrieval and navigation through collections easy and enjoyable. We provide text viewing possibilities, and graph manipulation operations such that the user would be able to explore the collection effectively and with minimal effort.

7 CONCLUSION AND OUTLOOK

The presented visualization approach supports historians in their work with collections of historical documents. This is achieved by extracting historically relevant information about the documents and by exploiting this information in order to determine potential relations between documents. We organize the data and present it graphically, in a way that can reveal patterns and connections between the documents which are otherwise difficult to spot.

It is important to stress that even though our system is dedicated to the historical domain, it can be relatively easily adapted to other domains. This can be done by selecting appropriate features that are specifically important for expressing inter-document similarity within these domains. The visualization method can be considered to be domain independent, as it represents objects according to similarity measurements regardless of the way these measurements were obtained.

Moreover, it is possible to think of other ways for determining domain important information. For example, one could utilize domain specific lexicons, and consider document terms that appear in such lexicons to be more important for the similarity measurements. A possible domain of application is for example law documents, where a lexicon of legal terms could be used for obtaining the inter-document similarities.

Many components in our pipeline can be improved or substituted with alternatives. For example, we intend to enhance the aliasing performance by incorporating knowledge about the structure of the NEs and their relative importance. A possible approach would be tagging NEs with labels such as *title*, *first name*, *last name*. This knowledge can then be integrated in the form of tree kernels that would further constrain the acceptable name variations. The feasibility of this direction is exemplified in [9]. We also intend to experiment with an off-the-shelf Coreference Resolution system such as BART [10] and compare the results of both approaches. Other sources for future experiments are additional layouts, similarity measures, clustering algorithms, and query expansion techniques.

We have currently carried out a relatively small scale and mostly qualitative evaluation of the usefulness of our visualization model. Designing a valid evaluation scheme in our setup is very challenging. The notions of similarity or relatedness are difficult to translate into clear evaluation schemes as the objects of analysis are full-length documents, and in particular rhetorical speeches that encompass a multitude of topics. Nevertheless, we plan to evaluate our setup in a more systematic manner in order to find out whether the similarity measurements according to NEs and general lexicon indeed correlate with human judgments of similarity and relatedness.

As our approach is user-oriented, one of the central aspects of our future evaluation scheme will be user evaluation and feedback. We plan to introduce our GUI tool to students and researchers in history departments, apply our tool on historical text databases they use and receive feedback on their experiences with the system.

We believe that evaluation and enhancement of the system in the proposed directions can further establish the relevance and usefulness of our approach.

ACKNOWLEDGEMENTS

We are thankful to Todd Shore for his involvement in this project. We would like to thank Caroline Sporleder for her dedicated guidance and valuable advice on the paper and to Martin Schreiber for his feedback on the system from the user perspective. The first author has been supported by an Erasmus Mundus scholarship in the framework of the EMMC "European Masters Program in Language and Communication Technologies (LCT)" (FPA 2007-0060/001 MUNB123). The second author has been supported by grant ME838 of the Czech Republic Ministry of Education, Youth and Sport.

REFERENCES

[1] Castro Speech Database. Retrieved 12 March 2010, from http://lanic.utexas.edu/la/cb/cuba/castro.html.

[2] C. Biemann, 'Chinese whispers: an efficient graph clustering algorithm and its application to natural language processing problems', in *TextGraphs '06: Proceedings of TextGraphs: the First Workshop on Graph Based Methods for Natural Language Processing*, pp. 73–80, Morristown, NJ, USA, (2006). Association for Computational Linguistics.

[3] K. Börner, C. Chen, and K.W. Boyack, 'Visualizing knowledge domains', *Annual review of information science and technology*, **37**(1), 179–255, (2003).

[4] J.R. Finkel, T. Grenager, and C. Manning, 'Incorporating non-local information into information extraction systems by Gibbs sampling', in *Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics*, pp. 363–370, Ann Arbor, MI, (2005). Association for Computational Linguistics.

[5] T. M. J. Fruchterman and E. M Reingold, 'Graph drawing by force-directed placement', *Software: Practice and Experience*, **21**(11), (1991).

[6] D. Nadeau and S. Sekine, 'A survey of named entity recognition and classification', in *Named Entities: Recognition, classification and use*, eds., D. Nadeau and S. Sekine, John Benjamins, Amsterdam and New York, (2009).

[7] R.W. Schvaneveldt, 'Pathfinder associative networks: studies in knowledge organization', *Ablex Series In Computational Science*, (1990).

[8] J. Shawe-Taylor and N. Cristianini, *Kernel methods for pattern analysis*, Cambridge University Press, Cambridge, England, 2004.

[9] Y. Versley, A. Moschitti, M. Poesio, and X. Yang, 'Coreference systems based on kernels methods', in *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, Manchester, England, (2008). Computational Linguistics.

[10] Y. Versley, S.P. Ponzetto, M. Poesio, V. Eidelman, A. Jern, J. Smith, X. Yang, and A. Moschitti, 'BART: A modular toolkit for coreference resolution', in *Proceedings of the 6th International Conference on Language Resources and Evaluation*, Marrakech, Morocco, (2008). LREC.

The Impact of Distributional Metrics in the Quality of Relational Triples

Hernani Costa[1] and **Hugo Gonçalo Oliveira**[2] and **Paulo Gomes**[3]

Abstract. This work analyses the benefits of applying metrics based on the occurrence of words and their neighbourhoods in documents to a set of relational triples automatically extracted from corpora. In our experimentation, we start by using a simple system to extract semantic triples from a corpus. Then, the same corpus is used for weighting each triple according to well-known distributional metrics. Finally, we take some conclusions on the correlation between the values given by the metrics and the evaluation made by humans.

1 INTRODUCTION

There has been a lot of work concerning the extraction of structured information from text in order to build or to enrich knowledge bases. This kind of work addresses both easing the access to information transmitted in natural language and also the reduction of the manual labour involved in the creation and maintenance of knowledge bases. Being more specific, lexical knowledge bases, such as Princeton WordNet [16], have revealed to be very useful in the achievement of tasks where understanding the meaning of natural language is critical, which go from machine translation to question answering or information retrieval (IR).

Most works on the automatic extraction of lexico-semantic knowledge from text are inspired by the work of Hearst [22], and rely on the identification of a set of textual patterns that frequently indicate semantic relations. However, the improvement of such methods and their evaluation is also an intensive process, especially when regarding unstructured text, where there are few boundaries and ambiguity is always a problem. One possible way to improve the precision of these methods automatically is to add a step to the extraction procedure, where information relating words is weighted according the words' distributional similarity. To achieve this, distributional metrics, typically used in IR, assign a value to pairs of terms according to their occurrences in a collection of documents.

The main goal of this work is to assess the benefits of applying distributional metrics to a set of semantic triples, automatically extracted from a newspaper corpus of Portuguese. Therefore, we have computed the later metrics for the output of a system which extracts relational triples. The extraction process is based on the identification of discriminating textual patterns for five types of semantic relations. A set of extracted triples was then manually evaluated by human judges and, in order to analyse to what extent distributional metrics could be considered in a filter, the former results were compared to the values of the later metrics.

[1] CISUC, University of Coimbra, Portugal, email: hpcosta@student.dei.uc.pt
[2] CISUC, University of Coimbra, Portugal, email: hroliv@dei.uc.pt
[3] CISUC, University of Coimbra, Portugal, email: pgomes@dei.uc.pt

This paper is organised as follows: in section 2, work on the automatic extraction of knowledge from text is presented; in section 3, well-known IR distributional metrics, used in our experimentation, are introduced; the goals of this research are stated in section 4; the experimentation performed and the obtained results are discussed in section 5; finally, before concluding, some additional related work is referred in section 6.

2 AUTOMATIC EXTRACTION OF KNOWLEDGE FROM TEXT

Information extraction (IE) [20] is the generic task of automatically extracting structured information from natural language inputs. Events or relationships, as well their arguments, are identified and then extracted in a structured representation, such as set of relational triples, $t = (e_1, r, e_2)$, where e_1 and e_2 are the arguments and are meant to denote entities, while r denotes a relationship between e_1 and e_2, or their meaning, in the case of a semantic relation. During the last decade, in opposition to using a fixed set of documents on a specific domain, the Web started to be seen as an infinite source of written data, suitable for IE. In this context, domain-independent systems like Snowball [1], KnowItAll [15], or more recently, TextRunner [14], have been developed to extract facts, concepts, and relationships from the Web.

Nevertheless, in our research we are more interested in lexico-semantic knowledge, which is knowledge about words of a language and their meanings, not specialised on any domain, and which can be used to create or enrich lexical resources such as WordNet [16], for English, or PAPEL [19], for Portuguese. Most of the work on the acquisition of lexico-semantic knowledge targets the automatic extraction of semantic relations such as hyponymy [22, 8, 9], part-of [3, 17], causality [18], as well the extraction of groups of similar words [28, 35, 26] or conclusions on words and their senses [13]. Work on the extraction of similar words is mostly based on statistics over the contexts surrounding the words. On the other hand, the extraction of semantic relations can combine the later methods with linguistic information, such as discriminating patterns that can be found in text to denote some relation. Besides introducing a set of patterns indicating the hyponymy relation, Hearst [22] presents a method for acquiring hyponymy relations from text, which inspired many works on relation extraction, not only for English, but also for other languages, including Portuguese (e.g. [2]). Furthermore, Hearst [22] presents an algorithm for the automatic discovery of less intuitive patterns. Briefly, this method uses several seed pairs of entities which are known to be related by a known relation type, and then searches, in a corpus, for sentences where both words of a pair occur. Although this method selects the most frequent indicating patterns,

Hearst patterns might not be frequent enough, especially in small collections, so other authors [10] propose using this kind of patterns on the Web in order to maximise the number of relations extracted.

Besides corpora text, semi-structured resources, such as dictionaries or encyclopedias, are also popular sources of lexico-semantic knowledge. The automatic extraction of knowledge from electronic dictionaries started during the 1970s [7], and continued through the 1980s and 1990s. However, despite being representative of a language, using simpler vocabulary and thus leading to less parsing issues, dictionaries are limited and static resources. Therefore, collections of unrestricted text seem to be a good alternative for automatic knowledge extraction, which can be used to enrich knowledge extracted from dictionaries or existing knowledge bases.

Also, the collaborative encyclopedia Wikipedia has recently been receiving more and more attention concerning IE, including the extraction of lexico-semantic knowledge (see for instance [29] [23]).

3 DISTRIBUTIONAL METRICS

Information retrieval (IR) [32] is the task of locating specific information within a collection of documents, or other natural language resources, according to some request. Among IR methods, we can find a large number of statistical approaches based on the occurrence of words in documents. Having in mind the distributional hypothesis [21], which assumes that similar words tend to occur in similar contexts, these methods are suitable, for instance, to find similar documents based on the words they contain or to compute the similarity of words based on their co-occurrence.

Here, we present some distributional metrics that can be found throughout the literature. In their expressions, e_i and e_j correspond to entities, which can be words or expressions, represented as strings of characters that are compared; $C = (d_1, d_2, d_3, ..., d_{|C|})$ is a collection of documents used to calculate the metrics; $P(e_i)$ is the number of documents ($d_n \in C$) where e_i occurs; and $P(e_1 \cap e_2)$ is the number of documents where e_i and e_j co-occur.

The measure of Cocitation, in expression 1, was first presented in [33] as a similarity measure between scientific papers, after analysing their references. However, it has been applied to other contexts like the similarity between Web pages [11]. In the original expression 1, $P(d_i \cap d_j)$ is the number of documents in the collection ($d_n \in C$) referring both documents d_i and d_j and $P(d_i \cup d_j)$ is the number of documents referring at least to one of the documents d_i and d_j.

$$Cocitation(d_i, d_j) = \frac{P(d_i \cap d_j)}{P(d_i \cup d_j)} \tag{1}$$

Still, in the scope of this work, we have adapted this expression to measure the similarity between textual entities, which results in expression 2, where $P(e_i \cap e_j)$ is the number of documents containing both entities e_i and e_j and $P(e_i \cup e_j)$ is the number of documents containing at least one of the entities. After this adaption, the measure of Cocitation can be seen as the *Jaccard* coefficient, used in statistics to compare similarity and diversity in two sets.

$$Cocitation(e_i, e_j) = \frac{P(e_i \cap e_j)}{P(e_i \cup e_j)} \tag{2}$$

Term Frequency - Inverse Document Frequency (TF-IDF) [30], in expression 3, is a popular measure in IR which weights (w) the relevance of a term (e_i) in a document (d_j), $w(e_i, d_j)$. Also, in the following expression, $f(e_i, d_j)$ is the frequency, or the number of times e_i occurs in d_j.

$$w(e_i, d_j) = (1 + \log_2 f(e_i, d_j)) * \log_2 \left(\frac{|C|}{P(e_i)} \right) \tag{3}$$

When measuring similarity between two objects, it is common to describe these objects as feature vectors which can be compared. Each entry of these vectors is a numerical aspect describing the object and, in the context of the similarity of documents or words, can be for instance the relevance of a word, or its occurrences in a context. Then, the simplest way to compare the vectors ($\vec{v}$ and $\vec{w}$) is to use the cosine similarity, presented in equation 4.

$$\cos(\vec{v}, \vec{w}) = \frac{\vec{v}.\vec{w}}{||\vec{v}||.||\vec{w}||} \tag{4}$$

Latent Semantic Analysis (LSA) [12] is a measure typically used to rank documents according their relevance to a query. It is based on the cosine similarity, which can be expanded into expression 5, to calculate the similarity between entities in a query and entities in the documents. For the sake of clarity, expression 5 considers a query with only two entities, however the query can consist of more than two entities. Using this measure, higher ranked documents, which have higher cosine values, are those containing entities more similar to the ones in the query. In the calculation of LSA, the weight of each entity in a document ($w(e_i, d_k)$ and $w(e_j, d_k)$) can be obtained using TF-IDF, the number of occurrences of e_i in d_k, or other method to compute the relevance of a word in a document.

$$Lsa(e_i, e_j) = \frac{\sum_{k=1}^{|C|} w(e_i, d_k).w(e_j, d_k)}{\sqrt{\sum_{k=1}^{|C|} w^2(e_i, d_k)} \sqrt{\sum_{k=1}^{|C|} w^2(e_j, d_k)}} \tag{5}$$

Lin [25] presents a theoretical discussion on the definition of similarity. He proposes a measure which does not assume any kind of domain model as long as it has a probabilistic model and is not defined directly by a formula. Still, the proposed measure is derived from a set of assumptions on similarity – the similarity between two objects is the ratio between the information common to both of the objects and the information needed to describe each one of them. Lin shows the generality of its measure when he applies it to domains that go from the similarity between ordinal values (e.g. *good*, *average*, *excellent*), to feature vectors or word similarity, as well as the calculation of semantic similarity based on a taxonomy. Expression 6 is Lin's measure applied to the similarity of two terms, based on their distribution in a corpus. There, the information common to both terms is given by the documents where they co-occur and the information needed to describe them is the sum of the documents where each term occurs.

$$Lin(e_i, e_j) = \frac{2 * \log P(e_i \cap e_j)}{\log P(e_i) + \log P(e_j)} \tag{6}$$

The algorithm called PMI-IR [35] uses Pointwise Mutual Information (PMI) and IR to measure the similarity of pairs of words. More precisely, PMI-IR was used to identify (near) synonym words based on their co-ocurrences in the Web, using expression 7, or variations of the later tuned for a specific search engine.

$$Pmi(e_i, e_j) = \log_2 \left(\frac{P(e_i \cap e_j)}{P(e_i) * P(e_j)} * |C| \right) \tag{7}$$

A completely different metric [24], based on the significance of the words in a corpus, was used to measure the similarity between two

words. In expression 8, which measures the significance of entity e_i, the number of occurrences of e_i in corpus C is given by $O(e_i, C) = \sum_{j=1}^{N} f(e_j, C)$. Expression 9 computes the similarity between entities e_i and e_j.

$$sim(e_i) = \frac{-\log\left(\frac{f(e_i,C)}{O(e_i,C)}\right)}{-\log\left(\frac{1}{O(e_i,C)}\right)} \quad (8)$$

$$\sigma(e_1, e_2) = sim(e_1) * sim(e_2) \quad (9)$$

4 RESEARCH GOALS

The most common ways to evaluate new knowledge involve either manual inspection by human judges or the comparison with a gold standard. While the former is time-consuming, tedious, hard to repeat and most of the times subjective to the judge's criteria, the latter is very dependent on the available resources.

Automatic methods have been developed to validate knowledge automatically extracted from text, based on existing resources (e.g. corpus, knowledge bases, the Web), which are usually exploited together with pre-conceived assumptions (e.g. related words tend to co-occur, some relation can be denoted by a set of discriminating textual patterns) and some mathematical formula to quantify the quality of the new knowledge. Additionally, in order to calculate the confidence on their results or to improve the precision of knowledge extraction systems, several authors (see section 6) have taken advantage of distributional metrics.

Having this in mind, the main goal of our work is to study how existing distributional metrics may be used to improve the quality of information extracted automatically from text and also how evaluation may benefit from using these metrics.

To study the impact of the later, we have integrated several metrics in a very basic system that aims the extraction of lexico-semantic knowledge from text. The system is based on a set of semantic grammars which include textual patterns that frequently denote semantic relations. We are aware that it captures an excess of extraneous and incorrect information, especially from unstructured text. However, regarding the goal of this work, this is not a problem but an added value, since we explicitly aim to test whether the metrics applied are capable of identifying these situations. Furthermore, using machine learning techniques, we will ascertain if it is possible to come up with a new metric based on one or several existing metrics.

5 EXPERIMENTATION

This section presents the experimentation carried out to study the possibility of using distributional metrics to improve the precision of relational triples, automatically extracted from Portuguese text.

5.1 Extraction approach

The extraction approach follows five stages that result in a set of relational triples, which will be used to study the metrics.

1. **Manual creation of the extraction grammars**: semantic grammars, based on frequent textual patterns, are manually created specifically for the extraction of the following kinds of semantic triples between entities[4]: synonymy (SINONIMO_DE), hypernymy (HIPERONIMO_DE), part_of (PARTE_DE), cause_of (CAUSADOR_DE) and purpose (FINALIDADE_DE). Some of the patterns in the grammars were confirmed after a pattern discovery algorithm [22] using relation instances from PAPEL [19], over the WPT05 corpus[5].
2. **Automatic extraction of the semantic relational triples**: each sentence of a textual input is analysed by a parser according to the semantic grammars and a triple set, $T = (t_1, t_2, ..., t_n)$, $t_i = (e_{i1}, r, e_{i2})$ is obtained.
3. **Lemmatisation of the arguments of the triples**: each entity argument of a triple, e_1 or e_2, is transformed into is lemma, if it is not already in that form. Multi-word entities have all its words lemmatised, which can sometimes lead to strange entities.
4. **Automatic removal of triples with stopwords in their arguments**: triples with at least one argument in a previously created stopwords list are removed from T.
5. **Additional extraction of triples**: additional hypernymy relations are extracted by analysing multi-word entities of the type *noun preposition noun*[6]. For each entity of this type a new hypernymy triple is extracted (see table 1). The new triples are added to T.

Table 1. Examples of triples extracted from multi-word entities.

Entity	New triple
casa_de_campo	*casa* HIPERONIMO_DE *casa_de_campo*
country_house	*house* HYPERNYM_OF *country_house*
garrafa_de_água	*garrafa* HIPERONIMO_DE *garrafa_de_água*
bottle_of_water	*bottle* HYPERNYM_OF *bottle_of_water*

5.2 Experimentation set-up

Through this experimentation we have used the part-of-speech annotated version of the CETEMPúblico corpus [31], provided by Linguateca[7], containing text from the newspaper Público, published between 1991 and 1998, and amounting to approximately 180 million words.

Due limitations on the processing time and storage resources we ended up using only the first 28,000 documents of CETEMPúblico, which contain 30,100 unique content words (considering only nouns, verbs and adjectives) and results in approximately 1 million of *word-in-document* relations.

A relational database, which can be seen as an occurrence matrix, was used to store this information and also the TF-IDF of all words. This occurrence matrix provides: (i) the number of documents, d_k; (ii) the number of times the word w_i occurs, (iii) the documents where w_i occurs; (iv) the number of words in d_k, N_{d_k}; (v) the total number of words in the corpus, N; and (vi) the relevance R_{w_i} of the word w_i in the corpus. With this information we can calculate the co-occurrence between w_1 and w_2 and the number of times both occur $P(w_1 \cap w_2)$.

[4] The grammars also defined each entity either as a simple word, or as words modified by generic adjectives (e.g. *bom*, *forte*; in English *good*, *strong*) or by prepositions (e.g. *de*, *com*; in English *of*, *with*).
[5] http://xldb.fc.ul.pt/wiki/WPT_05
[6] We have only used the preposition *de* or its contraction with an article: *de, do, da, dos, das*.
[7] http://www.linguateca.pt

5.3 Extraction results

For experimentation purposes, extraction was also performed over the first 50,000 documents of CETEMPúblico and a total amount of 20,308 triples was first obtained. Then, after the discarding phase, 5,844 triples (28.8%) were removed from the later set. Finally, additional extraction resulted in more 2,492 (17.2%) new triples.

The final triple set included 16,956 triples, more precisely 270 synonymy triples, 9,365 hypernymy, 1,373 part_of, 2,660 cause_of, and 3,288 purpose_of triples. Two example sentences and the triples extracted from them, as well as their translation, are shown in table 2. In the second example, one of the problems of the extraction system is in evidence: the parser can only connect the word *diplomat* with *Egypt* and not with the other countries in the enumeration, but an erroneous triple is extracted anyway.

Table 2. Extraction Examples of triples extracted from CETEMPúblico.

Sentence	Triple(s) extracted
... possibilidade de transplantar para o homem pulmões, rins ***ou outros*** *órgãos colhidos em porcos ...*	*órgão* HIPERONIMO_DE *pulmão* *órgão* HIPERONIMO_DE *rim*
... the possibility of transplanting to humans lungs, kidneys ***and other*** *organs obtained from pigs ...*	*organ* HYPERNYM_OF *lung* *organ* HYPERNYM_OF *kidney*
A delegação ***inclui*** *diplomatas do Egipto, Irão, Paquistão, Saudita e Senegal.*	*diplomata_do_Egipto* PARTE_DE *delegação* *Irão* PARTE_DE *delegaçcão* *Saudita* PARTE_DE *delegaçcão* *Senegal* PARTE_DE *delegaçcão*
The delegation includes diplomats from Egypt, Iran, Pakistan, Saudi Arabia and Senegal.	*diplomat_from_Egypt* PART_OF *delegation* *Iran* PART_OF *delegation* *Saudi_Arabia* PART_OF *delegation* *Senegal* PART_OF *delegation*

5.4 Application of the metrics

The distributional metrics referred in section 3, more precisely in expressions 2, 5, 6, 7 and 9, were implemented and normalised to fit the interval [0-100]. For instance, PMI-IR was normalised based on Bouma's [6] proposal. Also, calculation of the weights $w(e_i, d_k)$ in the LSA expression (5) was done by two different methods: the number of occurrences of entity e_i in the document d_k (LSA_o) and TF-IDF (LSA_t).

Each distributional metric was applied to the triple set, T, in the following manner: for each triple $t_i = (e_1, r, e_2), t_i \in T$, the distributional similarity between e_1 and e_2 was computed. For multi-word entities, the metrics were applied between each word of one entity and each word of the other, excluding stopwords, in order to calculate the average similarity value.

5.5 Manual evaluation

To evaluate the precision of the results, we selected random samples for each type of relation. The samples' sizes took the type of relation into consideration and were the following: 503 hypernymy triples (5.4%), 179 purpose relations (5.4%), 133 cause_of relations (5.0%), 71 part_of relations (5.2%) and of 270 synonymy relations (100%), totalling 1,156 triples, which were divided into ten random samples, each one evaluated by one of ten human judges.

Each human judge was asked to assign one of the following values to each triple, according to its quality:

- 0, if the triple was completely incorrect.
- 1, if the triple was not incorrect but something was missing in one or both of its arguments (e.g. a modifier) or the relation was very generic.
- 2, if the triple was correct.

A sentence describing the meaning of each relation was provided together with the triples to validate.

The results obtained for manual evaluation are presented in figure 1. As we can see, there are many incorrect triples, which show that the extraction system is far from perfect. Nevertheless, we were expecting to reduce the number of incorrect triples after applying a filter based on one or several distributional metrics.

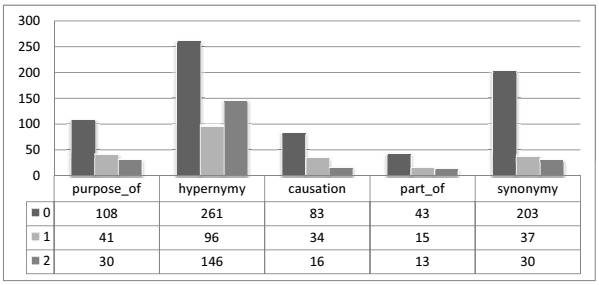


Figure 1. Manual Results

5.6 Manual evaluation vs. Distributional metrics

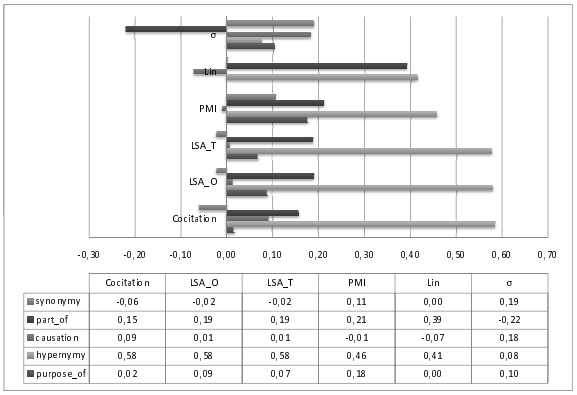
Table 3 shows some examples of extracted triples and puts side-by-side their manual evaluation and the calculated metrics. Since the triples were extracted from the same corpus used to obtain the metrics, the later values are never zero except for Lin's measure in the triple *palavra* HIPERONIMO_DE *beato*. However, this happens because these words only co-occur once and, nevertheless *palavra* is a very frequent word, *beato* is very infrequent.

In order to observe the relationships between the manual evaluation and the output values given by the metrics, the correlation coefficients between them were computed and are shown in figure 2. It is possible to observe that most metrics are strongly correlated with the quality of the triples, except for synonymy. This happens because all metrics except σ are based on co-occurrences and, in corpora text, synonymy entities, despite sharing very similar neighbourhoods, may not co-occur frequently in the same sentence [13] or even in the same document because they are alternative names for the same things. This might also be the reason for the low correlation coefficients with σ, which is based on the relevance of the terms.

Higher correlation coefficients are obtained for the hypernymy relation with the metrics of PMI and, especially, LSA and Cocitation/*Jaccard*, which suggests that hyponyms and their hypernyms tend to co-occur more frequently than causes or purposes. Also, there are more ways to denote the later relations in corpora text which led to less extracted and more incorrect triples. This is in conformity with an experience [19] where patterns denoting these relations were looked for in CETEMPúblico to validate semantic triples included in the lexical resource PAPEL[8]. On the other hand, part_of relations

[8] In that experience, only 4% and 10% of causes and purposes were re-

have good correlation coefficients with Lin's measure and LSA. Another conclusion is that, with this experience, the obtained values for LSA calculated with the occurrences of the entities (LSA_o) are very similar to the ones calculated with the TF-IDF (LSA_t). However, calculating the number of occurrences of a term in a document is much faster than computing the TF-IDF.



	Cocitation	LSA_O	LSA_T	PMI	Lin	σ
synonymy	-0,06	-0,02	-0,02	0,11	0,00	0,19
part_of	0,15	0,19	0,19	0,21	0,39	-0,22
causation	0,09	0,01	0,01	-0,01	-0,07	0,18
hypernymy	0,58	0,58	0,58	0,46	0,41	0,08
purpose_of	0,02	0,09	0,07	0,18	0,00	0,10

Figure 2. Correlation coefficients between manual evaluation and the distributional metrics

Furthermore, in order to study how the distributional metrics could be combined to create a new set of metrics, each one considering a different type of relation, we have used machine learning techniques, more specifically the toolkit Weka [37]. Several datasets were created, each one for a different relation. These datasets comprise a set of triple evaluation scores and their manual evaluation, as the entries of table 3, and were used for training several classification algorithms.

The best learned models using the algorithms of isotonic regression and also simple linear regression are shown in Table 4 together with their correlation coefficient. There are two situations where the models are not present because the obtained correlation coefficients were very low and it did not make sense to choose the best. As one can see, most of the best results for numeric classification were obtained with the isotonic regression model which picks the attribute that results in the lowest squared error, and defines several cut points, assuming a monotonic function.

Table 4. Learned metrics with higher correlation coefficient.

Relation	SimpleLinear	Corel	Isotonic	Corel
cause_of	(0.01*σ+0.05)	0.12	-	-
purpose_of	(0.02*Pmi-0.6)	0.22	Pmi	0.24
hypernym	(0.02*Cocitation+0.49)	0.56	Cocitation	0.66
part_of	(0.01*Lin+0.26)	0.28	Cocitation	0.38
synonymy	-	-	σ	0.22

The J48 was the best algorithm for discrete classification. J48 is an improved version of the C4.5 algorithm [27] and its result model is a decision tree, such as the one in figure 3, obtained using a 10-fold cross-validation test which classifies 59.1% of the purpose_of triples correctly. As one can see, this tree classifies the triples into one of the following classes, corresponding to the manual evaluation scores (0, 1 and 2).

spectively confirmed, against 18% and 22% of confirmed hypernymy and meronymy instances.

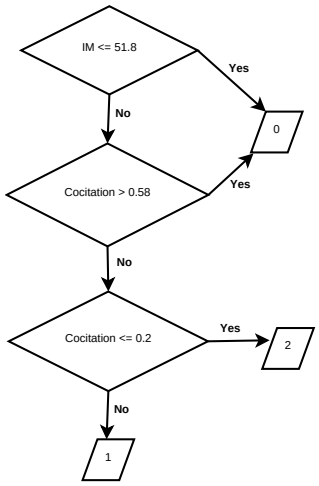


Figure 3. The J48 decision tree learned for purpose.

5.7 Additional experimentation

Based on the experimentation presented in the previous sections, we have analysed the impact of using a filter based on the best metrics obtained with the isotonic regression. Figures 4 and 5 present the evolution of the precision using different cut points on the Cocitation/*Jaccard* metric for the hypernymy and part_of triples. Of course that, while the cut point increases (leading to a gain in precision), less triples are obtained (at the cost of a slight loss of recall), but the majority of the discarded ones are wrong, leading to a higher precision. From a certain cut point, the amount of triples starts to decrease giving rise to more variations in the precision. Therefore, after observing figures 4 and 5 we would define 50 and 1 as adequate thresholds which would, for instance, filter the incorrect triple *theater* HYPERNYM_OF *to_refer*, presented in table 3.

Since other authors [8, 9, 36] propose computing LSA based on a term-term matrix $M(n, n)$, where n is the total number of terms and each entry, M_{ij} is the number of times terms i and j co-occur in a word context window, we made an additional experimentation using this kind of matrix and a context bounded by the beginning and the end of a sentence. We noticed that, in this specific experience, the LSA values and the correlation with the manual validation were not that different.

6 RELATED WORK

Methods for constructing lexical resources only by the identification of textual patterns, similar to Hearst [22], despite being recurrent, have several problems [1]. Many techniques have been proposed to improve them. For instance, taking advantage of other linguistic constructions (e.g. noun coordination) to improve extraction recall [28, 9]. Others [1, 34] propose improving recall and reducing the human effort by using a small set of seed instances or a few handcrafted extraction patterns to make the systems learn extraction patterns. An-

Table 3. Examples of extracted triples, their manual evaluation score and their computed distributional metrics.

Triple	Manual	Coc	LSA_o	LSA_t	PMI	Lin	σ
livro HIPERONIMO_DE *livro_de_reclamações* *book* HYPERNYM_OF *complaints_book*	2	100	100	100	100	94.85	27.5
nação SINONIMO_DE *povo* *nation* SYNONYM_OF *people*	2	4.21	7.92	8.21	66.65	55.12	35.79
violência CAUSADOR_DE *estrago* *violence* CAUSE_OF *damage*	2	1.60	4.38	4.47	63.90	29.51	43.82
palavra HIPERONIMO_DE *beato* *word* HYPERNYM_OF *pietist*	1	0.16	1.75	1.78	61.83	0	48.25
poder CAUSADOR_DE *algum_desigualdade* *power* CAUSE_OF *some_difference*	1	0.27	3.07	3.25	54.82	45.52	26.15
jogo FINALIDADE_DE *preparar* *game* PURPOSE_OF *prepare*	1	1.61	3.53	3.62	50.89	48.22	25.52
sofrer SINONIMO_DE *praticar* *suffer* SYNONYM_OF *practice*	0	0.73	1.34	1.37	52.04	27.77	34.25
atender FINALIDADE_DE *moderno* *answer* PURPOSE_OF *modern*	0	0.69	1.81	1.82	55.22	13.84	41.24
teatro HIPERONIMO_DE *referir* *theater* HYPERNYM_OF *to_refer*	0	0.58	1.31	1.27	46.92	38.57	24.48

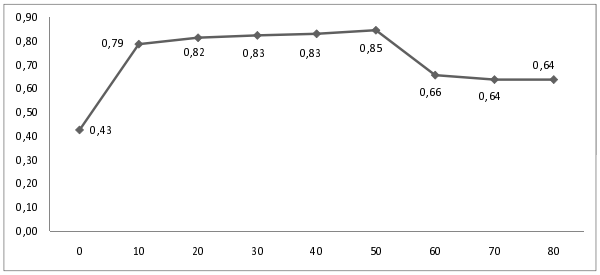


Figure 4. Evolution of the precision when increasing the threshold for the hypernymy relations.

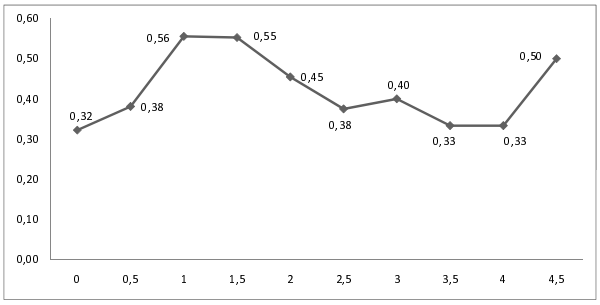


Figure 5. Evolution of the precision when increasing the threshold for the part_of relation.

other alternative is to use a huge corpus such as the Web (see for instance [1, 35, 15, 10]) to extract substantially more information.

However, these recall improvement measures tend to reduce the extraction precision. When it comes to improving the latter, distributional metrics are usually a good option to rank the triples based on the similarity between related entities.

While we have calculated all the metrics over a corpus, some of them can be adapted to target the Web and use the hits of a query on a search engine. The PMI-IR is an example of such a metric and some other metrics of this kind (e.g. WebJaccard, WebOverlap, WebDice) are presented in [5] and in [10].

As mentioned in section 3, PMI-IR was first developed to identify synonym words. For that specific task, it seemed to perform better than LSA (see [35]). KnowItAll [15] uses PMI-IR to compute the likelihood that some entity belongs to a class. PMI is calculated as the ratio between the search engine hit counts of each extracted instance and automatically generated indicative textual patterns (e.g. Hearst patterns) associated with the class.

Adaptations of LSA using a term-term matrix instead of a term-document have also been used to weight relational triples according to the distributional similarity of their arguments [8, 9, 36] which can be used to discard triples whose arguments are unlikely to be related.

Lin's similarity measure, adapted to measure the similarity between two synsets, is used in [26] to select the most suitable Wordnet synset for a group of related words extracted from text using a clustering algorithm.

Blohm et al. [4] study the impact of using several distributional measures (including PMI-IR) to reduce the noise in information extracted from the Web through pattern learning algorithms. In their experiments, an evaluation measure that considers the number of seed pairs of words that produced each learned pattern was the one which performed better. The later measure favoured more general patterns and penalised patterns which just held for a few examples.

Besides weights assigned according to distributional metrics, the number of times each triple was extracted is usually a good indicator not only of the correction of the triple, but also of its relevance. So, this hint is also used in several works ([15, 36]).

7 CONCLUSIONS AND FUTURE WORK

We have shown that the precision of systems capable of acquiring semantic knowledge from text may benefit from applying distributional metrics to their output. Although this work is made for Portuguese, we believe that it can be adapted to other languages with similar distributional behaviour.

If, on the one hand, it is possible to combine several metrics in a linear expression or in a decision tree, on the other hand, the best results were obtained using an isotonic regression that selected the metrics which minimised the squared error.

Most of the works similar to ours, but for English, propose using LSA-based filters. However, despite very close correlation results, for hypernymy and part_of relations, our adaptation of the Cocitation metric, which is basically the *Jaccard* coefficient, seems to be the most adequate for such a task. As several authors [9, 36] use LSA with a term-term matrix for performing tasks very close to ours, in the future, we are planning to further analyse the advantages and drawbacks of computing LSA by other means.

Furthermore, we will use more documents of the corpus to perform the same experiences presented in this paper, in order to observe the effect in the correlation coefficients. Also, we are planning to analyse the results of applying the resulting filters to semantic triples extracted from other textual sources or to assign weights to triples in existing resources for Portuguese, such as PAPEL.

REFERENCES

[1] E. Agichtein and L. Gravano, 'Snowball: Extracting relations from large plain-text collections', in *Proc. 5th ACM International Conference on Digital Libraries*, pp. 85–94, (2000).

[2] T. L. Baségio, *Uma Abordagem Semi-Automática para Identificação de Estruturas Ontológicas a partir de Textos na Língua Portuguesa do Brasil*, Ph.D. dissertation, Pontifícia Universidade Católica do Rio Grande do Sul PUCRS, 2007.

[3] M. Berland and E. Charniak, 'Finding parts in very large corpora', in *Proc. 37th Annual Meeting of the ACL on Computational Linguistics*, pp. 57–64, Morristown, NJ, USA, (1999). ACL.

[4] S. Blohm, P. Cimiano, and E. Stemle, 'Harvesting relations from the web: quantifiying the impact of filtering functions', in *Proc. 22nd National Conference on Artificial Intelligence (AAAI'07)*, pp. 1316–1321. AAAI Press, (2007).

[5] D. Bollegala, Y. Matsuo, and M. Ishizuka, 'Measuring semantic similarity between words using web search engines', in *Proc. 16th International conference on World Wide Web (WWW'07)*, pp. 757–766, New York, NY, USA, (2007). ACM.

[6] G. Bouma, 'Normalized (pointwise) mutual information in collocation extraction', in *Proc. Biennial GSCL Conference 2009, Meaning: Processing Texts Automatically*, pp. 31–40, Tbingen, Gunter Narr Verlag, (2009).

[7] N. Calzolari, L. Pecchia, and A. Zampolli, 'Working on the italian machine dictionary: a semantic approach', in *Proc. 5th Conference on Computational Linguistics*, pp. 49–52, Morristown, NJ, USA, (1973). ACL.

[8] S. A. Caraballo, 'Automatic construction of a hypernym-labeled noun hierarchy from text', in *Proc. 37th Annual Meeting of the Association for Computational Linguistics (ACL'99)*, pp. 120–126, Morristown, NJ, USA, (1999). ACL.

[9] S. Cederberg and D. Widdows, 'Using LSA and noun coordination information to improve the precision and recall of automatic hyponymy extraction', in *Proc. CoNLL*, pp. 111–118, (2003).

[10] P. Cimiano and J. Wenderoth, 'Automatic acquisition of ranked qualia structures from the web', in *Proc. 45th Annual Meeting of the Association of Computational Linguistics (ACL'07)*, pp. 888–895, Prague, Czech Republic, (June 2007). ACL.

[11] M. Cristo, E. S. de Moura, and N. Ziviani, 'Link information as a similarity measure in web classification', in *Proc. 10th Symposium On String Processing and Information Retrieval*, pp. 43–55. Springer, (2003).

[12] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, 'Indexing by latent semantic analysis', *Journal of the American Society for Information Science*, **41**, 391–407, (1990).

[13] B. Dorow, *A Graph Model for Words and their Meanings*, Ph.D. dissertation, Institut fur Maschinelle Sprachverarbeitung der Universitat Stuttgart, 2006.

[14] O. Etzioni, M. Banko, S. Soderland, and D. S. Weld, 'Open information extraction from the web', *Communications of the ACM*, **51**(12), 68–74, (2008).

[15] O. Etzioni, M. Cafarella, D. Downey, S. Kok, A. M. Popescu, T. Shaked, S. Soderland, D. S. Weld, and A. Yates, 'Web-scale information extraction in knowitall: (preliminary results)', in *Proc. 13th International Conference on World Wide Web (WWW)*, pp. 100–110, New York, NY, USA, (2004). ACM.

[16] *WordNet: An Electronic Lexical Database (Language, Speech, and Communication)*, ed., C. Fellbaum, The MIT Press, May 1998.

[17] R. Girju, A. Badulescu, and D. Moldovan, 'Automatic discovery of part-whole relations', *Computational Linguistics*, **32**(1), 83–135, (2006).

[18] R. Girju and D. Moldovan, 'Text mining for causal relations', in *Proc. FLAIRS Conference*, pp. 360–364, (2002).

[19] H. Gonçalo Oliveira, D. Santos, and P. Gomes, 'Relations extracted from a portuguese dictionary: results and first evaluation', in *Local Proc. 14th Portuguese Conference on Artificial Intelligence (EPIA 2009)*, (12-15 October 2009).

[20] R. Grishman, 'Information extraction: Techniques and challenges', in *International Summer School on Information Extraction (SCIE)*, pp. 10–27, (1997).

[21] Z. Harris, 'Distributional structure', in *Papers in Structural and Transformational Linguistics*, 775–794, D. Reidel Publishing Company, Dordrecht, Holland, (1970).

[22] M. A. Hearst, 'Automatic acquisition of hyponyms from large text corpora', in *Proc. 14th conference on Computational Linguistics*, pp. 539–545, Morristown, NJ, USA, (1992). ACL.

[23] A. Herbelot and A. Copestake, 'Acquiring ontological relationships from wikipedia using RMRS', in *Proc. ISWC 2006 Workshop on Web Content Mining with Human Language Technologies*, (2006).

[24] H. Kozima and T. Furugori, 'Similarity between words computed by spreading activation on an english dictionary', in *Proc. 6th Conference on European Chapter of the Association for Computational Linguistics (EACL)*, pp. 232–239, Morristown, NJ, USA, (1993). ACL.

[25] D. Lin, 'An information-theoretic definition of similarity', in *Proc. 15th International Conference on Machine Learning*, pp. 296–304. Morgan Kaufmann, (1998).

[26] P. Pantel and D. Lin, *Discovering word senses from text*, 613–619, ACM, New York, NY, USA, 2002.

[27] J. R. Quinlan, *C4.5: programs for machine learning*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

[28] B. Roark and E. Charniak, 'Noun-phrase co-occurrence statistics for semiautomatic semantic lexicon construction', in *Proc. 17th International Conference on Computational linguistics*, pp. 1110–1116, Morristown, NJ, USA, (1998). ACL.

[29] M. Ruiz-Casado, E. Alfonseca, and P. Castells, 'Automatising the learning of lexical patterns: An application to the enrichment of wordnet by extracting semantic relationships from wikipedia', *Data & Knowledge Engineering*, **61**(3), 484–499, (2007).

[30] G. Salton and C. Buckley, 'Term-weighting approaches in automatic text retrieval', *Information Processing and Management: an International Journal*, **24**(5), 513–523, (1988).

[31] D. Santos and P. Rocha, 'Evaluating CETEMPúblico, a free resource for portuguese', in *Proc. 39th Annual Meeting on Association for Computational Linguistics (ACL'01)*, pp. 450–457, Morristown, NJ, USA, (2001). ACL.

[32] A. Singhal, 'Modern information retrieval: A brief overview', *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, **24**(4), 35–42, (2001).

[33] H. Small, 'Co-citation in the scientific literature: A new measure of the relationship between two documents', *Journal of the American Society for Information Science*, **24**(4), 265–269, (1973).

[34] R. Snow, D. Jurafsky, and A. Y. Ng, 'Learning syntactic patterns for automatic hypernym discovery', in *Advances in Neural Information Processing Systems*, eds., Lawrence K. Saul, Yair Weiss, and Léon Bottou, 1297–1304, MIT Press, Cambridge, MA, (2005).

[35] P. D. Turney, 'Mining the web for synonyms: PMI–IR versus LSA on TOEFL', in *Proc. 12th European Conference on Machine Learning (ECML-2001)*, eds., Luc De Raedt and Peter Flach, volume 2167, pp. 491–502. Springer, (2001).

[36] T. Wandmacher, E. Ovchinnikova, U. Krumnack, and H. Dittmann, 'Extraction, evaluation and integration of lexical-semantic relations for the automated construction of a lexical ontology', in *Proc. 3rd Australasian Ontology Workshop (AOW 2007)*, eds., Thomas Meyer and Abhaya C. Nayak, volume 85 of *CRPIT*, pp. 61–69, Gold Coast, Australia, (2007). ACS.

[37] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*, Morgan Kaufmann, San Francisco, 1999.

Automatic Annotation of Media Field Recordings

Eric Auer, Peter Wittenburg, Han Sloetjes[1], Oliver Schreer, Stefano Masneri[2], Daniel Schneider, Sebastian Tschöpel[3]

Abstract. In the paper we describe a new attempt to come to automatic detectors processing real scene audio-video streams that can be used by researchers world-wide to speed up their annotation and analysis work. Typically these recordings are taken in field and experimental situations mostly with bad quality and only little corpora preventing to use standard stochastic pattern recognition techniques. Audio/video processing components are taken out of the expert lab and are integrated in easy-to-use interactive frameworks so that the researcher can easily start them with modified parameters and can check the usefulness of the created annotations. Finally a variety of detectors may have been used yielding a lattice of annotations. A flexible search engine allows finding combinations of patterns opening completely new analysis and theorization possibilities for the researchers who until were required to do all annotations manually and who did not have any help in pre-segmenting lengthy media recordings.

1 BACKGROUND

Many researchers in linguistics such as field workers and child language researchers have to work with real scenario sound and video material. Field recordings are often more challenging to process than lab recordings, for example for pattern recognition tasks. The reasons for this are manifold such as inadequate and varying position of the sensor devices (microphone, camera), various types of background noise, the need to use consumer grade devices etc. Standard speech and image recognition techniques only deliver very poor results for such recordings. Of course there are also many resources with better recording quality, but they often involve non-standard languages, long stretches of silence or regular patterns resulting from experimental settings etc. Yet, annotators would like to use any help they can get to make their work more efficient, because manual annotation is so time consuming.

There is often little knowledge about the analyzed languages, so we miss formal descriptions such as proper language models. The consequence is that researchers who want to analyze this sort of material need to first carry out manual annotations based on time consuming listening and watching. In 2008, we made statistics amongst 18 teams documenting endangered languages within the DoBeS[4] program to find out how much time is required for the most essential workflow steps. According to these statistics creating a transcription costs 35 times real-time (i.e. a transcription of an one-hour video requires at least 35 hours), a translation into a major language 25 times real-time and for any special linguistic encoding such as morphosyntactic glossing or gesture annotation the costs in general are much higher than 100 times real time.

Because annotating is so time-consuming, an increasing number of recordings in the archives of the Max Planck Institutes are not annotated and even not touched any more, i.e. valuable material cannot be included in analysis of the linguistic system, theoretical considerations and cultural and cognitive studies. Advanced annotation and analysis tools such as ELAN[5] and ANNEX[6] can facilitate the difficult work and can speed up the process slightly although no quantitative factors can be given. Yet these tools do not operate at the content level of the media streams.

2 DESIGN CONSIDERATIONS

Motivated by this unsatisfying development some brainstorming between researchers and technologists of two Max Planck Institutes on the one side and sound and image processing specialists from two Fraunhofer Institutes was initiated to discuss ways out leading to a three year innovation project funded by Max Planck Gesellschaft and Fraunhofer Gesellschaft. Actually an old idea spelled out in the Hearsay II system [1] was brought into consideration again. In Hearsay II more or less complex independent knowledge components were operating on the speech signals each of them writing their findings on a blackboard. Other knowledge components were added that analyzed the blackboard findings to finally create an automatic transcription of what was said. Such a knowledge based architecture has the potential of being used to let the user interact with the low level audio and video analysis components, which was one of the major requirements of the researchers at the Max Planck Institutes participating in this innovation project.

In AVATecH[7], detector components analyze audio or video input streams and generate annotations or intermediate results. Detectors can use the output of other detectors as input, in addition to the audio and video source files.

After having analyzed a preliminary evaluation corpus with a variety of recordings provided by the Max Planck Institutes, we found that the characteristics of the data are indeed challenging for acoustic analysis. 55 scenes from about 30 files include wind noise and similar, about 10 have reverb, about 15 considerable background noise (engines, people, etc.) and 5 contain humming sounds. About 20 scenes seem to be not useful for any type of audio analysis. The speech quality itself is also varying from “indistinguishable talking” to intelligible speech. The results of acoustic segmentation, speech

[1] Max Planck Institute for Psycholinguistics, Nijmegen, the Netherlands, email:eric.auer@mpi.nl
[2] Fraunhofer Heinrich Hertz Institute, Berlin, Germany
[3] Fraunhofer Institut für intelligente Analyse- und Informationssysteme IAIS, Sankt Augustin, Germany
[4] DoBeS: Dokumentation Bedrohter Sprachen, www.mpi.nl/dobes/
[5] ELAN: Eudico Linguistic Annotator, www.lat-mpi.eu/tools/elan/
[6] Annex: Web-based annotation viewer, www.lat-mpi.eu/tools/annex/
[7] AVATecH: Advancing Video Audio Technology in Humanities, see www.mpi.nl/avatech

detection, speaker clustering and gender detection with standard algorithms optimized for broadcast data were rather disappointing as was expected. Due to the variety of languages, classic mono-lingual speech recognition could not be applied.

The initial corpus analysis resulted in a number of conclusions:

- return to the blackboard type of scenario where "detectors of various sorts" will create annotations on a new specific tier
- start experimenting with so-called low hanging fruits, i.e. simple detectors that can be integrated quickly based on existing algorithms
- have smart search and filtering methods to allow researchers to easily browse through (complex) annotation lattices
- allow the researcher to interact with the annotations and easily modify parameters controlling the functioning of the detectors so that manual tuning can be used instead of using a "one size fits all" stochastic method
- rely on existing technologies where possible with respect to the annotation and search framework and the pattern detectors

3 ANNOTATION AND SEARCH FRAMEWORK

ELAN is currently one of the most widely used media annotation tools in various linguistic sub-disciplines and beyond. It allows researchers to hook up an arbitrary number of annotation tiers referencing custom vocabularies to multiple media streams that share the same timeline. The fact that annotations cannot only be attached to a time segment but also to annotations on other tiers provides support for the creation of complex annotation structures, such as hierarchical annotations trees.

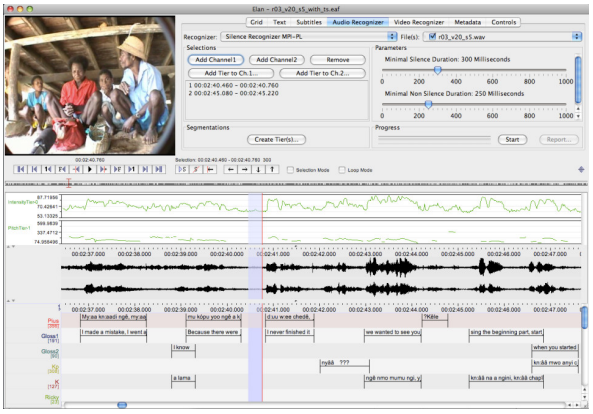


Figure 1. Use of a silence detector in ELAN 3.6: Detector parameters can be adjusted on the right side. A video viewer is on the left side. Results will be added as a new tier to the annotations and waveforms at the bottom.

In contrast to comparable tools [2] such as Frameline 47, ANVIL, EXMARaLDA or Advene[8], ELAN's advantages include an open-source core, unlike the commercial Frameline 47 or closed-source ANVIL. This is ideal for extending the tool with detection algorithms. Also, ELAN already supports numerous import and export formats (in contrast to EXMARaLDA or Advene) relevant for linguistic research such as PRAAT, CHILDES Chat[9], Shoebox[10] or Transcriber[11] data. Like most of the tools mentioned, ELAN is platform independent: It is available for Mac OS, Windows and Linux.

The underlying EAF (ELAN Annotation Format) schema emerged from the early discussions about models such as Annotation Graph [3] and it is flexible enough to cater for a large number of tiers with variable vocabularies being created by a number of (small) detectors. The screenshot in figure 1 depicts a typical ELAN window layout. ELAN has many functions including the possibility to start the well-known PRAAT[12] speech analysis software for a specific, detailed acoustic analysis.

ELAN is accompanied by TROVA[13], a flexible search engine that allows users to search for complex annotation patterns within annotation tiers, across several annotation tiers, over time and/or annotation sequences. Each pattern can be specified as a regular expression offering a large degree of flexibility. TROVA operates not only on the visualized resource, but can be used to operate on a whole selection of resources resulting from metadata searches or composed by the user. Using indexes created at resource upload or integration time, TROVA can operate very fast on large amounts of data. While the user reads the first results, TROVA continues to search further matches in the background when searching in a large corpus.

The current tools are an excellent starting point for improvements in the direction of adding new semi-automatic annotation and extended search functionality. Also, users are already familiar with the user interfaces, making it easy for them to adopt new functionality.

4 FIRST INTEGRATION EXAMPLE

The first recognition component that was integrated as a test case offers simple detection of pauses (silences) in sound recordings – in fact a well-studied detection problem, the potential errors of which are known. The user can configure the essential parameters in a graphical user interface and can inspect the results in a timeline view immediately after the execution. If necessary, the user can adjust parameters and try again. This feature of ELAN is already applied by a variety of users and it speeds up their work considerably. Some of the scenarios are:

- In experiment result analysis, users want to quickly index or remove periods of silence in order to reduce the length of the sound wave to be analysed to a minimum.
- Field linguists want to use the "annotation step through" function of ELAN to quickly navigate from one sequence of speech to the next, thus carrying out a first very rough selection of the material.
- Gesture researchers can now more easily create statistics that interrelate the timing of gesture and speech segments.

It is not solely the complexity of the detection function that counts: In this particular low-hanging fruit example it is the packaging into a tool such as ELAN and the convenient graphical interaction that are attractive to researchers. The typical errors produced by such detectors are in general not dramatic, since the researchers likely use the detected segments either just for quick inspection or as a base segmentation that might be manually corrected and extended. A complete API for plug-ins or components being executed on remote servers has been worked out and has been verified. This API is documented in a manual which is available online [4].

[8] Frameline 47: www.frameline.tv/ ANVIL: www.anvil-software.de/ EXMARaLDA: www.exmaralda.org/ Advene: http://liris.cnrs.fr/advene/

[9] CHILDES: http://childes.psy.cmu.edu/

[10] www.sil.org/computing/**shoebox**/ & www.sil.org/computing/**toolbox**/

[11] Transcriber: http://trans.sourceforge.net/

[12] PRAAT: www.fon.hum.uva.nl/praat/

[13] Annex & Trova: www.lat-mpi.eu/tools/annex/ or select "annotation content search" in the IMDI Browser at http://corpus1.mpi.nl/

5 LOW HANGING FRUIT DETECTORS

Currently a number of such low-hanging fruit detectors have been studied on test corpora and are being integrated into the ELAN framework. For audio signals we are working on robust audio segmentation, speech detection, speaker clustering and pitch contour detection. For video, we are working on the integration of shot and sub-shot boundary detection, motion recognition (camera and scene motion), face detection and tracking of body parts. We also investigate possibilities for gesture recognition.

5.1 Segmentation

Noise-robust segmentation of the audio stream into homogeneous segments inserts boundaries e.g. between speakers or at other significant acoustic changes. The algorithm will be capable of providing fine-grained segmentation of speaker utterances [5]. The user can control the granularity of segmentation by tuning a corresponding feedback parameter.

5.2 Speech detection

This detector finds audio segments which contain human speech, in a language-independent way. Naturally, weak audio quality is a drawback for the detection quality. Furthermore the various research recordings are very heterogeneous. Thus, we enable the user to manually annotate a small amount (less than five minutes) of non-speech segments in order to adapt the model to the given data which leads to a more robust detection.

5.3 Speaker clustering

A language-independent intra-document speaker clustering algorithm labels identical speakers within a single document with the same ID (see [6], [7] and [8]). The results can be used for removing the interviewer in a recording, or for extracting material from specific speakers from a recorded discussion. For optimization of the detection performance we use manual user input, e.g., the number of speakers or speaker audio samples.

5.4 Vowel and pitch contour detection

The pitch contour detector can allow researchers to graphically specify typical pitch contours and search for similar patterns. We already implemented a detector which tags vowel segments in audio recordings and annotates the corresponding time-spans with pitch and intensity properties such as for example minimum, maximum, initial or final f0 frequency, or volume. The detector invokes PRAAT to calculate f0 and volume curves of the input over time. Those are then used to find characteristic segments and annotate them.

5.5 Shot and sub-shot boundary detection

The shot and sub-shot boundary detector (see [9], [10] and [11]) identifies scene changes as well as considerable changes in the video scene. Since different shots refer to different camera operations, all the subsequent detectors work on a shot basis. Each detected scene as well as scene changes are marked by a still frame, in order to represent all of the content in the video and allow the user to browse through it without actually watching the video. The detector processes about 80 frames every second on a single core 3.6 GHz Pentium IV, i.e. an hour of video is processed in less than 20 minutes. An example of the results from this detector is shown in figure 2.

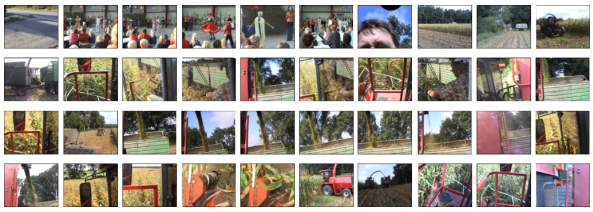


Figure 2. This figure shows the results of a shot and sub-shot boundary detection. At well-defined moments, a frame is taken to give a quick overview of what is happening in a video, allowing e.g. quick navigation.

5.6 Motion recognition

The motion recognizer detects either motion of the camera (pan, tilt or zoom) or motion inside the scene (see [12] and [13]). This is particularly useful in case the user wants to distinguish between static or dynamic shots, or wants to know when and where a change in the background occurs. The results of the motion recognizer can also be used by other detector to compensate the effect of the camera motion while tracking objects or people inside of a scene. The detector processes about 25 frames per second.

5.7 Face detection

The face recognition detector, based on the Viola-Jones algorithm (see [14] and [15]), is used to identify the number of persons in a scene. The detector can be configured to find frontal faces, profile faces or both, and has also limited face tracking capabilities. The speed of the detector depends on the parameter set, but can reach 40 frames per second.

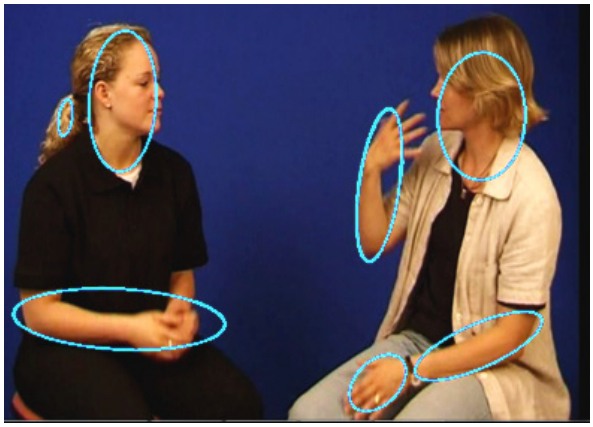


Figure 3. This figure shows the results of body part tracking, based on a previous skin-color detection step, for one frame. Note that no explicit body model is used by this detector at the moment.

5.8 Body part tracking

This detector identifies body parts (hands, arms and heads) and then tracks them. It estimates at first the skin colour (see [16] and [17]) for each shot in the video and then identifies and tracks the different body parts, which are then approximated by ellipses. By tracking the body parts the user knows when movements/signs begin or finish,

when hands join, what is the position of the hands with respect to other body parts. This detector runs at about 50 frames per second. An example of the results from this detector is shown in figure 3: Tracked body parts are marked with ellipses. Note that the detector does not yet have a body model, but tracks moving skin-color areas.

5.9 Gesture recognition

The gesture recognition tool identifies simple hand gestures, still or moving. This detector is still in early development and neither qualitative nor quantitative tests have been made. ELAN is already used for manual annotation of sign language (see [18] and [19]), but machine support could help to improve speed and quality of the annotation process a lot.

5.10 Robustness and user interface

Currently, we are testing the behaviour of the existing detectors with respect to the variety of material we have in our 800 GB test corpus (300 GB of audio and 500 GB of video, mostly WAV and MPEG 1, 2 and 4). It is obvious that we need to study, how we can create simple to use interfaces to allow users to influence detection parameters easily and to immediately see the effects. Moreover we would like to gather feedback from users in an iterative process to improve the quality of the analysis.

Using a common interface, detectors in AVATecH can be called either from ELAN or from a custom batch processing tool which we called ABAX (AVATecH Batch Executor). For that, each of the detectors comes with a metadata file which specifies the necessary parameters and input and output files to call that detector. While the metadata can define choice lists and the ranges for numerical parameters, it does not attempt to be a machine readable representation of the parameter semantics. Instead, it contains a short description of each item for use in human user interfaces which can be automatically generated from the metadata. Note that all parameters must have defaults: This helps the users to quickly get first results. Once they found a detector to be useful for their annotation work, they can adjust settings (for a group of input files or separately for each file) to improve the quality of the results.

Detectors can be made available for a number of operating systems, using a platform independent design for communication: Parameters, file names and log / progress information are sent through pipelines as plain (XML) text. This even allows the use of (intranet) "detector servers" by sending the pipelines through a TCP/IP connection. Caller and detector still have to share a (network) filesystem for media and (XML or CSV) result files. A direct Java API is also available, for cases where the focus is on tight integration.

6 SUMMARY

With integrating a number of detection components that create layers of annotations that can be easily used by ELAN and TROVA, we are making a new step in facilitating the work of manual annotators. Also, coarse automated annotations can help to find useful recordings in unexplored corpora. As has been seen from the very simple silence detector, which we used as first example, it can speed up the work of researchers by factors when the interaction interface is simple and the user can stay in a well-understood tool framework. A set of first low hanging fruit detectors has been tested and is being integrated into the ELAN framework. The results will be analyzed to determine which other more complex detectors will be added and how user interaction options need to be modified to maintain attractiveness for researchers who are not only interested in pure recognition scores but also want to understand underlying mechanisms.

ACKNOWLEDGEMENTS

AVATecH is a joint project of Max Planck and Fraunhofer, started in 2009 and funded by MPG and FhG.

REFERENCES

[1] L.D. Erman, F. Hayes-Roth, V.R. Lesser, et al., 'The Hearsay-II speech-understanding system: Integrating knowledge to resolve uncertainty', *ACM Computing Surveys (CSUR)*, **12**(2), 213–253, (1980).

[2] K. Rohlfing, D. Loehr, S. Duncan, et al., 'Comparison of multimodal annotation tools - workshop report', *Gesprächsforschung - Online-Zeitschrift zur verbalen Interaktion*, **7**, 99–123, (2006).

[3] S. Bird and M. Liberman, 'A formal framework for linguistic annotation (revised version)', *CoRR*, **cs.CL/0010033**, (2000).

[4] E. Auer, H. Sloetjes, and P. Wittenburg, *AVATecH Component Interface Specification Manual*, http://www.mpi.nl/research/research-projects/language-archiving-technology/avatech/, (2010).

[5] Shih-Sian Cheng, Hsin-Min Wang, and Hsin-Chia Fu, 'BIC-based speaker segmentation using divide-and-conquer strategies with application to speaker diarization', *IEEE transactions on audio, speech, and language processing*, **18**(1), 141–157, (2010).

[6] K. Biatov and J. Köhler, 'Improvement speaker clustering using global similarity features', in *Proceedings of the Ninth International Conference on Spoken Language Processing*, (2006).

[7] K. Biatov and M. Larson, 'Speaker clustering via bayesian information criterion using a global similarity constraint', in *Proceedings of the Tenth International Conference SPEECH and COMPUTER*, (2005).

[8] D.A. Reynolds, 'Speaker verification using adapted gaussian mixture models', *Speech Communication Journal*, **17**(1-2), (1995).

[9] C. Petersohn, 'Fraunhofer HHI at TRECVID 2004: Shot boundary detection system', in *TREC Video Retrieval Evaluation Online Proceedings*. TRECVID, (2004).

[10] C. Petersohn, 'Sub-shots - basic units of video', in *EURASIP Conference Focused on Speech and Image Processing, Multimedia Communications and Services*, Maribor, Slovenia, (2007).

[11] J. Boreczky and L. Rowe, 'Comparison of video shot boundary detection techniques', *Journal of Electronic Imaging*, **5**(2), 122–128, (1996).

[12] N. Atzpadin, N. Kauff, and O. Schreer, 'Stereo analysis by hybrid recursive matching for real-time immersive video conferencing', *Trans. on Circuits and Systems for Video Technology, Special Issue on Immersive Telecommunications*, **14**(3), 321–334, (2004).

[13] A. Dumitras and B.G. Haskell, 'A look-ahead method for pan and zoom detection in video sequences using block-based motion vectors in polar coordinates', in *Proceedings of the 2004 International Symposium on Circuits and Systems*, volume 3, pp. 853–856, (2004).

[14] P. Viola and M. Jones, 'Robust real-time object detection', in *Second International Workshop on Statistical and Computational Theories of Vision*, (2001).

[15] P. Viola and M. Jones, 'Rapid object detection using a boosted cascade of simple features', in *Conference on Computer Video and Pattern Recognition*, (2001).

[16] S. Askar, Y. Kondratyuk, K. Elazouzi, P. Kauff, and O. Schreer, 'Vision-based skin-colour segmentation of moving hands for real-time applications', in *Proceedings of the 1st European Conference on Visual Media Production (CVMP 2004)*, London, United Kingdom, (2004).

[17] S. Masneri, O. Schreer, D. Schneider, S. Tschöpel, R. Bardeli, et al., 'Towards semi-automatic annotation of video and audio corpora', in *Seventh international conference on Language Resources and Evaluation (LREC)*, (2010).

[18] H. Lausberg and H. Sloetjes, 'Coding gestural behavior with the NEUROGES-ELAN system', *Behavior Research Methods, Instruments, Computers*, **41**(3), 841–849, (2009).

[19] O. Crasborn, H. Sloetjes, E. Auer, and P. Wittenburg, 'Combining video and numeric data in the analysis of sign languages with the ELAN annotation software', in *Proceedings of the 2nd Workshop on the Representation and Processing of Sign languages: Lexicographic matters and didactic scenarios*, ed., C. Vetoori, pp. 82–87, Paris, (2006). ELRA.

Proppian Content Descriptors in an Augmented Annotation Schema for Fairy Tales

Thierry Declerck[1], **Antonia Scheidel**[2], **Piroska Lendvai**[3]

Abstract. This paper describes a proposal for combining linguistic and domain specific annotation for supporting Cultural Heritage and Digital Humanities research, exemplified in the fairy tale domain. Our goal is to semi-automatically annotate fairy tales, in particular to locate and mark up fairy tale characters and the actions they are involved in, which can be subsequently queried in a corpus by both linguists and specialists in the field. The characters and actions are defined in Propp's structural analysis to folk tales, which we aim to implement in a fully fledged way, contrary to existing resources. We argue that the approach devises a means for linguistic processing of folk tale texts in order to support their automated semantic annotation in terms of narrative units and functions.

1 INTRODUCTION

Various theories in narratology research may assign properties to characters in different ways. For example, in actant theory (cf. the actant model, developed by [2], or [3] for more details) actants are positions, kind of behavioral patterns, in a story situation. Importantly, one and the same actor can serve as a different actant in different situations – as opposed to the classical view of possessing consistent roles throughout a story, advocated for example by folklorist Vladimir Propp (see [6]). According to Propp, main characters (or *dramatis personae*) that are occuring in a fairy tale may be the following[4]:

1. Hero: a character that seeks something;
2. Villain: opposes or actively blocks the hero's quest;
3. Donor: provides the hero with an object of magical properties;
4. Dispatcher: sends the hero on his/her quest via a message;
5. False Hero: disrupts the hero's success by making false claims;
6. Helper: aids the hero;
7. Princess: acts as the reward for the hero and the object of the villain's plots;
8. Her Father: acts to reward the hero for his effort.

Additionally to the characters, Propp introduces the following concepts or units for the interpretation of Russian fairy tales:

31 Functions At the heart of the *Morphology of the Folktale* (see [6]) lies the description of actions that can be performed by the dramatis personae of a folktale. These so-called *functions* are the prototypical invariant features of fairy tales such as "Conflict", "Call for Help", "Kidnapping", "Test of Hero", and so on. Functions are frequently divided into sub-functions: in the case of function *A: Villainy*, they range from A^1*: The villain abducts a person* to A^{19}*: The villain declares war*. Functions and subfunctions are described in detail and illustrated with examples from Russian folktales in [6].

A sequence of all the functions from one folktale is called a *scheme* and can be used as a formal representation of the tale (see Fig. 1 for an example).

$$\gamma^1\beta^1\delta^1A^1C\uparrow\{[DE^neg.Fneg.]^3d^7E^7F^9\}G^4K^1\downarrow [Pr^1D^1E^1F^9 = Rs^4]^3$$

Figure 1. Functional scheme for *The Magic Swan-Geese*

150 Elements. In Appendix I of *Morphology of the Folktale*, Propp provides what he calls a "list of all the elements of the fairy tale". The list contains 150 elements, distributed over six tables:

1. Initial Situation
2. Preparatory Section
3. Complication
4. Donors
5. From the Entry of the Helper to the End of the First Move
6. Beginning of the Second Move

Some of the 150 elements appear alone, others are grouped under a descriptive heading. If these "element clusters" (as shown in Fig. 2) are counted as one, the appendix contains 56 - as they shall tentatively be called in the following - narratemes. About a third of the narratemes can be mapped directly to functions, such as the aforementioned *30-32. Violation of an interdiction*. Other narratemes can be combined to form an equivalent to a function (together, narratemes *71-77: Donors* and *78: Preparation for the transmission of a magical agent* can presumably be considered as a superset to the information expressed by function *D: First Function of the donor*.

Another group of narratemes, however, goes beyond the 31 functions: *70. Journey from home to the donor*, for example, can be seen as filling the gap between the functions ↑*: Departure* and *D: First Function of the donor*. The first table (*Initial Situation*[5]) contains a

[1] DFKI GmbH, Language Technology Lab, Germany, declerck@dfki.de
[2] DFKI GmbH, Language Technology Lab, Germany, Antonia.Scheidel@dfki.de
[3] Research Institute for Linguistics, Hungarian Academy of Sciences, piroska@nytud.hu
[4] Source: http://www.adamranson.plus.com/Propp.htm

[5] Propp makes use of the symbol α*: Initial Situation* to refer to everything that happens before the hero's parents announce their departure, but it is not a function as such.

```
30-32. Violation of an interdiction
      30. person performing
      31. form of violation
      32. motivation
```

Figure 2. Example for a narrateme

multitude of narratemes dedicated to the circumstances of the hero's birth and other events/situations which precede the actual adventure.

Furthermore, Table 1 (Initial Situation) includes two "element-clusters" describing the hero and false hero, respectively, in term of 'future hero' (see Fig. 3).

```
10-15. The future hero
      10. nomenclature; sex
      11. rapid growth
      12. connection with hearth, ashes
      13. spiritual qualities
      14. mischievousness
      15. other qualities
```

Figure 3. Example for an element cluster serving as profile

A closer examination of the appendix reveals such profiles for each of the dramatis personae, although sometimes spread over two clusters or narratemes.

The longer term objective of our work consists in devising a means for linguistic processing of folk tale texts in order to support their automated semantic annotation in terms of Propp's theory, using an appropriate encoding schema, but which could be adapted to other theories of fairy tales or literary genres. As a starting point for our work, we analysed available resources that could be used or re-used in our work, and make a among others a first proposal for an augmented XML annotation schema.

2 RESOURCES

Similarly to the well-developed application of NLP technologies to certain domains (e.g. financial news, biomedicine), it is equally important albeit less trivial in Digital Humanities to identify a set of textual as well as domain-specific semantic resources based on which complex information can be gained and modeled.

2.1 Textual resources

About 200 fairy tales are available from the Afanas'ev collection (see [1]) that is English translations of the Russian originals, on which Propp based his model. Additionally, popular tales such as "Little Red Riding Hood" exist in many versions in many languages. We focus in this paper on one of the German versions of this tale, to illustrate our proposed combination of linguistic annotation and Proppian character functions. This annotation exercise is part of the D-SPIN project[6] and a first prototype of an automated annotation of a multilingual version of this tale is part of a use case of the CLARIN project[7]. The annotation exercise is planned to be extended to all Grimm tales, as they become available within the Gutenberg project[8].

[6] http://weblicht.sfs.uni-tuebingen.de/englisch/index.shtml
[7] http://www.clarin.eu/external/
[8] See http://www.gutenberg.org/

2.2 Semantic resources

There exist a number of computational models for processing, annotation, and generation of fairy tales, in the form of semantic resources or annotation schemas. These describe narration in terms of moves and their ingredients, often drawing on the work of Propp. We consider two ontologies that model certain aspects of Propp: ProppOnto ([5]) and [8][9], built for generation and control purposes in interactive digital storytelling and games. It is notable that such resources typically do not specify or type which linguistic elements need to be associated with the model's constituents (e.g. concepts, relations) in order to express a domain-specific function in natural language. In other words, the semantic information is exclusively encoded in the ontology classes, with no link to potential linguistic realizations, allowing little or no flexibility of reusing the resource across languages or across domains.

A central part of the MONNET project[10] consists of modeling linguistic information in ontologies; we are extending the default domains of application of MONNET (i.e. financial reporting, eGovernment) to Digital Humanities.

2.3 Annotation schemas

Additionally, there are a few XML-based annotation schemas for Proppian functions we are aware of: the Proppian Fairy Tale Markup Language (PftML)[11], and one supporting the generation of animated movies, based on Proppian functions [7].

While the first schema is used for analysis purposes, it remains at a very coarse-grained level, allowing inline textual markup, which is typically assigned at the sentence or paragraph level. Association of specific linguistic expressions with the functions is not supported. The second schema pertains to word level, but only of (unstandardized) semantic features that integrate generic semantic roles (agent, location, etc.) and Proppian functions; linguistic information is not considered at all. Neither of the schemas support cross-referencing of objects (i.e., roles and actions) in the text, which would be a desirable property of resources in Humanities research. More detailed description on our former work with PftML and ProppOnto is given in [4].

3 AUGMENTING THE PftML SCHEMA AND THE SEMANTIC RESOURCES

In our investigation of the available resources for our work, we noticed that many of these incorporate only a subset of the elements described by Propp, depending on the application at hand. So for example PftML is concentrating on the functions but does not adress the elements. Therefore, one of our first steps consisted of creating an augmented annotation schema involving all elements of fairy tales that were discussed by Propp, as described above.

Below we give a preliminary instantation of the augmented annotation schema, which we call APftML (Augmented Proppian fairy-tale Markup Language), currently under development. APftML is intended to afford a fine-grained, stand-off annotation of folk (or fairy)

[9] See http://www.fdi.ucm.es/profesor/fpeinado/projects/kiids/apps/protopropp/ respectively http://eprints.aktors.org/440/01/tuffieldetal.pdf
[10] MONNET – Multilingual Ontologies for Networked Knowledge – is an R&D project co-funded by the European Commission, with grant 248458. See http://cordis.europa.eu/fp7/ict/language-technologies/project-monnet_en.html
[11] developed by S. Malec, see http://clover.slavic.pitt.edu/sam/propp/theory/propp.html

tales in accordance with concepts introduced in Vladimir Propps Morphology of the Folktale. It is loosely based on PftML. APftML will integrate / interact with ProppOnto.

```
<annotation>
  <InitialSituation>
  <Content>Es war einmal eine kleine
  suesse Dirne, die hatte jedermann lieb,
  der sie nur ansah, am allerliebsten
  aber ihre Grossmutter, die wusste gar nicht,
  was sie alles dem Kinde geben sollte.
  Einmal schenkte sie ihm ein Kaeppchen
  von rotem Sammet, und weil ihm das
  so wohl stand und es nichts anders
  mehr tragen wollte, hiess es nur
  das Rotkaeppchen.</Content>
  </InitialSituation>
...
</annotation>
```

In the given example just above, we introduce with "Content" element an explicit way of encoding the exact portion of the text that is interpreted as describing the "InitialSituation"[12].

```
<Narrateme>
  <Command subtype="command" id="i0">
     Eines Tages sprach seine Mutter zu ihm:
     Komm, Rotkaeppchen, da hast du ein Stueck
     Kuchen und eine Flasche Wein,
     bring das der Grossmutter hinaus;
     sie ist krank und schwach und
     wird sich daran laben.</Command>
  <Agent id="p0">seine Mutter</Agent>
  <Patient id="p1">Rotkaeppchen</Patient>
  <Content>bring das der Grossmutter hinaus
  </Content>
  <Motivation>sie ist krank und schwach und
    wird sich daran laben</Motivation>
</Narrateme>
```

In this second example, the segment of the text containing the "Command" is given again by the "Content" element. This annotation also introduces semantic information on "agent" and "patient" and we provide for an index for the function "Command", so that we can refer to it if the text we encounter a textual segment. Those new elements and features are major additions to PftML, and include also Proppian elements that were not consistently used in the previously existing ontologies.

4 TOWARDS A STANDARDIZED TEXTUAL AND LINGUISTIC ANNOTATION

For the annotation of the textual and linguistic information we suggest to use the family of standards developed within TEI (Text Encoding Initiatitive)[13] and ISO TC 37/SC4[14], also in order to verify the potential of those standards for serving as pivot format for the representation of textual and linguistic information. As a first step we apply the TEI encoding standard, so that we get clearly marked textual content objects. We distinguish here between the TEI header and the text itself, as the two examples below show:

```
<TEI xmlns="http://www.tei-c.org/ns/1.0"
 xmlns:ht="http://www.w3.org/1999/xhtml">
 <teiHeader>
 <fileDesc>
- <titleStmt>
  <title>Rotkaeppchen</title>
- <respStmt>
  <resp>collector</resp>
  <persName>Gebrueder Grimm</persName>
  </respStmt>
  </titleStmt>
- <publicationStmt>
  <p>"http://gutenberg.spiegel.de/?i
     d=5&xid=969&kapitel=230&cHash=
     b2042df08brotk#gb_found"</p>
  </publicationStmt>
  <sourceDesc />
  </fileDesc>
- <revisionDesc>
  <change when="2010-06-16">Tentative
    Annotation</change>
  </revisionDesc>
  </teiHeader>

<body>
- <p>
  ...
  <w xml:id="t3">Es</w>
  <w xml:id="t4">war</w>
  <w xml:id="t5">einmal</w>
  <w xml:id="t6">eine</w>
  <w xml:id="t7">kleine</w>
  <w xml:id="t8">suesse</w>
  <w xml:id="t9">Dirne</w>
  <w xml:id="t10">,</w>
  <w xml:id="t11">die</w>
  <w xml:id="t12">hatte</w>
  <w xml:id="t13">jedermann</w>
  <w xml:id="t14">lieb</w>
  <w xml:id="t15">,</w>
  <w xml:id="t16">der</w>
  <w xml:id="t17">sie</w>
  <w xml:id="t18">nur</w>
  <w xml:id="t19">ansah</w>
  <w xml:id="t20">,</w>
  ...
 </body>
```

The TEI encoding of the body is including the information about the tokenization of the text, and this is an anchor point for the subsequent annotation levels, using a stand-off strategy: all the annotation levels can point back to the original text on the base of the numbering of the tokens.

4.1 Morpho-syntactic annotation

On the top of TEI annotation we are applying the ISO-MAF standard for morpho-syntactic annotation [15], linking its elements to the words

[12] We do not provide for a translation here, since we assume the story to be well known.
[13] http://www.tei-c.org/index.xml
[14] http://www.tc37sc4.org/
[15] see http://pauillac.inria.fr/∼clerger/MAF/html/body.1_div.5.html

as they are marked by TEI, using the token IDs we introduced into the 'w' elements:

```
<?xml version="1.0" encoding="UTF-8"?>
<maf:MAF xmlns:maf="__">
<maf:tagset>
<dcs local="KON" registered=
"http://www.isocat.org/datcat/DC-1262"
rel="eq"/>
<!-- __ -->
</maf:tagset>
<maf:wordForm tokens="t135">
<fs>
<f name="lemma"><symbol value="sehen"/></f>
<f name="partOfSpeech"><symbol value="VVIMP"/>
 </f>
<f name="grammaticalNumber"><symbol value=
 "singular"/></f>
</fs>
...
```

This specific morpho-syntactic annotation, using an XML representation of feature structures, is pointing to the token number 135, which in the text is the verb "see" in a particular grammatical form (i.e. the imperfect). We are currently working on the syntactic annotation following the guidelines of ISO SynAF[16]. This annotation is building on the top of MAF and is annotating (groups of) words with constituency (e.g. nominal or verbal phrases, etc.) and dependency information (subject, object, predicate etc.). The idea is that an identified subject of a sentence can usually be mapped onto the "Agent" element of a Proppian function. But here we have to take into account also the modus of the sentence (Active vs. Passive modus).

5 INTEGRATION WITH THE APftML ANNOTATION SCHEMA

This step is straightforward: we take the functional annotation and add the proper indexing information, so that all the descriptors of the functional annotation are linked to the available levels of linguistic annotation. This can look like:

```
<semantic_propp>
<Command subtype="Interdiction" id="Command1"
inv_id="Violated1" from="t135" to="t148">
</semantic_propp>
```

T135 and t148 are used here as defining a region of the text for which the Propp function holds. Navigating through the different types of IDs included in the multi-layered linguistic annotation, the user can extract all kind of (possibly) relevant linguistic information and combine it with the functional annotation in terms of Propp.

On the basis of this combination of distinct types of annotation – linguistic and (Proppian) functional–, we expect that the work of finding variations in fairy tales can be enhanced, but most significant is probably the fact that it is getting much easier to pursue text-based research on tales.

6 CONCLUSIONS

We described ongoing work in adapating and augmenting existing annotation schemas of fairy tales. We described also a strategy for using natural language processing in order to support the automated markup of character roles and functions. Generalizing the results will shed light on the computational applicability of a manually created Humanities resource, in our case of Propp's method for narratives, in Digital Humanities research. If we can detect genre-specific narrative units on evidence based on a linguistically annotated corpus, we plan to take this research further and analyse higher level motifs (such as narratemes), as well as other types of narratives.

[16] http://www.iso.org/iso/catalogue_detail.htm?csnumber=7329

ACKNOWLEDGEMENTS

The ongoing work described in this paper has been partially supported by the European FP7 Project "MONNET" (Multilingual Ontologies for Networked Knowledge), with Grant 248458, and by the BMBF project "D-SPIN"[17]. Investigating higher-order content units such as motifs is the focus of the AMICUS project[18], which is supported by The Netherlands Organisation for Scientific Research (NWO).

REFERENCES

[1] A. Afanas'ev, *Russian fairy tales*, Pantheon Books, New York, 1945.
[2] A. J. Greimas, *Sémantique structurale*, Larousse, Paris, 1966.
[3] D. Herman, 'Pragmatic constraints on narrative processing: Actants and anaphora resolution in a corpus of North Carolina ghost stories', *Journal of Pragmatics*, **32(7)**, 959–1001, (2000).
[4] P. Lendvai, T. Declerck, S. Darányi, P. Gervás, R. Hervás, S. Malec, and F. Peinado, 'Integration of linguistic markup into semantic models of folk narratives: The fairy tale use case', in *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, ed., N. Calzolari et al., Valletta, Malta, (May 2010). European Language Resources Association (ELRA).
[5] F. Peinado, P. Gervás, and B. Diaz-Agudo, 'A description logic ontology for fairy tale generation', in *Proceedings of LREC*. ELRA, (2004).
[6] V.J. Propp, *Morphology of the folktale*, University of Texas Press:, Austin, 1968.
[7] N. Takahashi, D. Ramamonjisoa, and O. Takashi, 'A tool for supporting an animated movie making based on writing stories in xml', in *Proceedings of IADIS International Conference Applied Computing*, (2007).
[8] M. M. Tuffield, D. E. Millard, and N. R. Shadbolt, 'Ontological approaches to modelling narrative.', in *Proceedings of the 2nd AKT DTA Symposium*. Aberdenn University, (1 2006).

[17] http://weblicht.sfs.uni-tuebingen.de/englisch/index.shtml
[18] http://amicus.uvt.nl

Adapting Standard NLP Tools and Resources to the Processing of Ritual Descriptions

Nils Reiter[1] and **Oliver Hellwig**[2] and **Anand Mishra**[2] and **Irina Gossmann**[1]
and **Borayin Maitreya Larios**[2] and **Julio Rodrigues**[1] and **Britta Zeller**[1] and **Anette Frank**[1]

Abstract. In this paper we investigate the use of standard natural language processing (NLP) tools and annotation methods for processing linguistic data from ritual science. The work is embedded in an interdisciplinary project that addresses the study of the structure and variance of rituals, as investigated in ritual science, under a new perspective: by applying empirical and quantitative computational linguistic analysis techniques to ritual descriptions. We present motivation and prospects of such a computational approach to ritual structure research and sketch the overall project research plan. In particular, we motivate the choice of frame semantics as a theoretical framework for the structural analysis of rituals. We discuss the special characteristics of the textual data and especially focus on the question of how standard NLP methods, resources and tools can be adapted to the new domain.

1 INTRODUCTION

Led by the observation of similarities and variances in rituals across times and cultures, ritual scientists are discussing the existence of a "ritual grammar", an abstract underlying – and possibly universal – structure of rituals, which nevertheless is subject to variation. It is controversial whether such structures exist, and if so, whether they are culture-independent or not.

Our interdisciplinary project[3] addresses this issue in a novel empirical fashion. Using computational linguistics methods, we aim at obtaining quantitative analyses of similarities and variances in ritual descriptions, thereby offering ritual scientists new views on their data.

Ritual researchers analyze rituals as complex event sequences, involving designated participants, objects, places and times. Such sequences are usually encoded in natural language descriptions. However, the knowledge of recurrent structures in ritual event sequences is often private among researchers devoted to particular cultures or scientific fields, because an all-encompassing theoretical framework for the analysis of rituals across different cultures does not yet exist. In our work, we attempt to make characteristic properties and structures in rituals overt. For this sake, we apply formal and quantitative computational linguistic analysis techniques on textual ritual descriptions. We will investigate data-driven approaches to detect regularities and variations of rituals, based on semi-automatic semantic annotation of ritual descriptions, thereby addressing this research issue in a novel empirical fashion.

As a ritual can be divided into complex event sequences, the computational linguistic analysis of ritual descriptions needs to focus on discourse semantic aspects: the recognition and analysis of events and roles, temporal relations between events and coreference and anaphora resolution regarding participants of these events, to name just a few. In order to capture variations and similarities across rituals, it is important to analyze and quantify variations in event successions (e.g., is a specific action accompanied by another one, or strictly followed by it?), as well as variance regarding the ontological type of participants (what kinds of materials or living beings are subject to or involved in specific actions in different roles?).

Computational Linguistics resources and tools for the analysis of ritual structure. Computational Linguistics has developed a variety of resources and processing tools for semantic and discourse processing that can be put to use for such a task. The community has developed semantic lexica and processing tools for the formal analysis of events and their predicate-argument structure, in terms of semantic roles [11, 16, 24], temporal relation recognition [32], and anaphora and coreference resolution [29, 33, 23]. Using these resources and processing tools, we can compute structured and normalized semantic representations of event sequences from textual descriptions of rituals, and thus identify recurrent patterns and variations across rituals by quantitative analysis. Frame semantics [11], with its concept of scenario frames connected by frame relations and role inheritance, offers a particularly powerful framework for the modeling of complex event sequences. It can be used to structure event sequences into more abstract concepts that may subsume different kinds of initial, transitional or closing events of rituals. Through the annotation of word senses, using lexical ontologies such as WordNet [10], we can observe and analyze variations in the selectional characteristics of specific events and their roles across rituals. The integration of corpora and ontologies [2] offers possibilities to reason over corpora and external knowledge resources.

Processing ritual descriptions with standard NLP tools. The semantic annotations, however, need to be built upon linguistically preprocessed data. This preprocessing consists of several layers, starting with tokenization, part of speech tagging, and shallow or full syntactic analysis. Semantic analysis tasks, such as semantic role labeling or coreference resolution typically builds on these pre-processing levels. As a basis for semantic annotation we use existing open-source systems for tokenizing, part of speech tagging, chunking or parsing. Automatic anaphora and coreference resolution provide im-

[1] Department of Computational Linguistics, Heidelberg University, Germany
[2] South Asia Institute, Heidelberg University, Germany
[3] The project is part of a collaborative research center (Sonderforschungsbereich, SFB) on "Ritual Dynamics". Over 90 researchers from 21 scientific fields work the structure and dynamics within and across different cultures. http://www.ritualdynamik.de

portant information for a coherent textual representation based on semantic role analysis. The systems we use for this preprocessing are data-driven, and have proven to obtain high performance scores, as they are typically trained on large corpora. In fact, such statistical systems often outperform rule-based systems.

However, there is one caveat: Most currently available training (and testing) data is taken from the news domain or encyclopedias like Wikipedia, which represent one or more particular domain(s). The assumption that data-driven approaches can be applied to an arbitrary new domain relies on the availability of training data for this domain. This, however, is rarely the case, especially if we move to "small" domains featuring special linguistic phenomena combined with restricted textual sources and a complete lack of annotated textual material.

In this paper, we report on first steps to provide a proof of concept for using computational linguistic resources and analysis methods for the study of ritual structures, based on small collections of data, analyzed at all intended levels of representation. In particular, we present initial studies that assess (i) the performance of standard NLP tools and resources for processing linguistic data from the ritual domain and (ii) the need and basic methods for domain adaptation.

Section 2 presents the project research plan and related work. In section 3, we discuss special linguistic characteristics of the textual data that have an impact for automatic processing. Section 4 presents experiments that measure the performance of standard NLP processing tools on various linguistic depths and assess basic domain adaptation techniques. Section 5 presents our methodology for performing the semantic annotation of ritual descriptions by assessing the coverage of existing resources, the need for domain adaptation as well as a principled work flow to enable future automation. Finally, we present an outlook on the type of analyses we expect to produce to enable empirical studies of the structure and variance of rituals. Section 6 describes plans for future work and concludes.

2 COMPUTATIONAL LINGUISTICS FOR RITUAL STRUCTURE RESEARCH

2.1 Project research plan

The project is divided into two consecutive stages of research, which concentrate on corpus creation and annotation and on the analysis and exploitation of the data, respectively.

Corpus creation and annotation. In the first stage, a comprehensive corpus of linguistically and semantically annotated rituals from different cultures will be created from natural language descriptions of rituals that are procured by experts. The semantic annotation follows the frame semantics paradigm [11] and comprises both general linguistic and ritual-specific annotations.

As we aim at an empirical basis for the conceptualization of the domain, we automatically identify relevant domain terms on the basis of scientific publications on ritual research which in turn can serve to establish a base vocabulary for the annotation with ritual-specific concepts.

Analyzing the structure of rituals. Based on the semantic annotation of ritual descriptions, logical and statistical methods will be deployed to detect recurring structures in ritual descriptions, as well as systematic variances. In close cooperation with the ritual researchers, we will provide tools for the exploration of our data-driven, quantitative analyses of rituals.

2.2 Related work

Central to the structure of rituals are sequences of events and participants involved in these events. Hence, an important research topic is the **detection and analysis of event chains** in texts. The use of frame semantics as a useful abstraction layer for analyzing event chains has been investigated in [1]. A case study demonstrated how relations between instances of frames and roles can be inferred in context, using frame relations as well as contextual information, such as coreference or syntactic association. A related shared task on "linking roles in discourse" [27] is being organized as part of SemEval 2010. Recently, a statistical approach has been proposed for unsupervised detection of event chains, using co-occurrence of a single discourse entity as argument of different verbs as well as coreference information as criteria for extracting event chains [3, 4]. Results of applying similar linguistic and computational techniques to a corpus of Sanskrit texts are reported in [13], where chains of events are used to detect the temporal structure of a corpus.

Another central issue related to our work is **domain adaptation**, because most NLP tools are trained on news corpora. An interesting approach for addressing the domain adaptation problem is augmenting the feature space to model both domain and general, domain-independent characteristics [6]. A very similar approach employs hierarchical bayesian prior to encourage the features to take similar weights across domains, unless the differences of the data demand otherwise [12]. Both methods make use of labelled data. A contrastive approach has used an instance weighting framework, where unlabeled instances of the target domain contribute to the model estimations [15].

3 RITUAL DESCRIPTIONS

We collect ritual descriptions from different sources. The collection process has been started with Hindu rituals from Nepal and rituals from the Middle East, but we plan to extend it to rituals from Ancient Egypt and the Middle Ages in central Europe. All our methods and techniques are culture-independent and can be adapted to other, non-English, languages.

We decided to concentrate on translated ritual descriptions that have already been published in scientific literature in order to quickly collect larger amounts of data that is relevant and trustworthy. All ritual descriptions are entered by a ritual researcher. We use a trac Wiki[4] as an interface, because it (i) allows easy to follow structuring rules, (ii) is readable by every project member on the Web without knowledge of XML or other markup languages and (iii) is designed for automatic processing.

In the following, we discuss specific properties of the ritual descriptions in our corpus that are relevant from a computational linguistics point of view.

3.1 Textual sources

We use two types of textual sources. The first comprises studies by ritual researchers that deal with the religious, ethnologic and social background of rituals and are strongly theory-oriented. These texts will serve as a basis for building a ritual specific ontology, starting from a common terminology [26]. The second type of texts are descriptions of rituals. These sources form the basis of the ritual corpus and are, therefore, of special importance for the project. Two subtypes of ritual descriptions can be distinguished.

[4] `http://trac.edgewall.org`

Ethnographic observations are an important source for our knowledge of how rituals are performed in modern times. These texts are written in English, though not always by native speakers. Some scholars tend to intersperse purely descriptive passages with theoretical interpretations of what was observed. The actual course of the rituals can thus not always be separated clearly from personal interpretations (see 3.2.5).

Translations of indigenous **ritual manuals** that may date back several centuries are the second subtype of the ritual descriptions. Originally, the manuals are written in non-English languages (e.g., Sanskrit), but English translations of them have been published in ethographic literature. Contrary to the ethnographic observations, these sources are mainly prescriptive in character. Since many of these manuals are intended as a kind of memory aid for ritual practitioners, they often record only the most important or extraordinary steps of a ritual, while typical, recurrent elements are omitted. This selective choice of content complicates the alignment of such manuals with the exhaustive descriptions of modern observers.

The subtype of ritual descriptions is stored as meta data attached to the source text, along with the bibliographic source of the descriptions, original language and related types of information.

3.2 Text characteristics

Dealing with ritual descriptions requires handling of special phenomena on the lexical, syntactical and discourse-level. We describe these challenges in the following.

3.2.1 Foreign terms

A ritual description produced by a ritual expert (be it a researcher or a practitioner) often contains terminology specific to the cultural context of the ritual. In most cases, English counterparts for these terms do not exist. Therefore, they often remain untranslated in the texts (although transliterated into Latin characters).

(1) He sweeps the place for the sacrificial fire with *kuśa*.

Kuśa is a Sanskrit term for a kind of grass that is very important in Vedic rituals. For this ritual, it is necessary to sweep with *kuśa* and not any other grass.

The term *kuśa* has never been seen by a common, newspaper trained part of speech tagger nor is it contained in a lexicon of a rule-based grammar. We therefore decided to annotate such terms with English paraphrases as in Example 2. For automatic processing, the original terms are replaced by the paraphrases and are later re-inserted.

(2) He sweeps the place for the sacrificial fire with <grass * kuśa>.

3.2.2 Fixed expressions

Most rituals contain fixed expressions consisting of multiple words or sentences. These expressions are often prescribed pieces of text which have to be spoken or chanted while a ritual is performed (e.g., *Our father* in Christian church service).

(3) Salutation to Kubera reciting the mantra *arddha-māsāḥ [. . .]*;

There is no common way in handbooks or scientific literature to refer to such fixed expressions. Sometimes, prayers or chants have a title or name; sometimes, first words or the refrain are given and the expert knows the exact expression.

As most fixed expressions cannot be translated literally, we adopt them as unanalyzed expressions in a foreign language. We ask the ritual experts to mark them as such, so that we can replace them with indexed placeholders during processing and re-insert them later.

3.2.3 Imperatives

As ritual manuals are often written by and for ritual practicioners, they contain a high amount of imperative sentences. In a randomly selected sample of ritual descriptions, we found 20% of the sentences realized in an imperative construction. The ritual description with the highest amount of imperatives contains over 70% of sentences with imperative constructions. By contrast, in the British National Corpus, only about 2 % of the sentences contain imperatives.

3.2.4 PP-attachments and nested sentences

Prepositional phrases (PPs) are quite common in the data, as becomes apparent in Example 1. This introduces ambiguities that are hard to resolve. Deeply embedded PPs (4) are difficult to attach correctly, but appear in the texts regularly.

(4) . . . worship of the doors of the house of the worshipper.

The frequency of syntactic coordination and nested sentence structures is varying between languages and text types. In Sanskrit, which is the source language of most of our data, long and nested sentences are very common. This characteristic is also reflected in the texts' translations into English, as the translators (i) try to preserve the original text character as much as possible and (ii) do not aim at producing well-to-read English sentences.

The joint occurrence of PP attachment, coordinations and sentence embedding are a challenge for syntactic processing. Example 5 illustrates the interaction of coordination (*italic*) and PP attachments (underlined) in a long sentence.

(5) Beyond the members of the lineage, these visits lead to the paternal aunts of three generations which includes father's *and* grandfather's paternal aunts *and* their daughters *and* granddaughters, the maternal uncles *and* maternal aunts of their grandmother *as well as* their maternal uncles of three generations.

This leads to a combinatorial explosion of possible analyses and – in case of statistical disambiguation – a parser is deemed to make wrong guesses. Therefore, since full-fledged syntactic analyses are not necessarily needed for role semantic labeling (see e.g. [9]), we opted for flat syntactic analysis based on chunks.

3.2.5 Interpretations

Ritual descriptions that have been published in scientific literature often do not contain "clean" descriptions restricted to the ritual performance only. Instead, the description of a ritual performance is interwoven with comments or interpretations that help the reader to understand the ritual.

(6) The involvement of the nephews can be understood as a symbolic action to address those of the following generation who do not belong to the lineage of the deceased.

Example 6 is clearly not an event that occurs during the ritual, but a scientific interpretation. Although it is in principle possible to annotate such sentences with frames and frame elements, they represent a different level of information that does not belong to the ritual itself. As we want to automatically extract common event sequences from the ritual descriptions, such interpretations need to be clearly separated from descriptions of factual events.

In order to systematically address this issue, we divided the sentences into three classes:

1. Sentences that clearly indicate events happening during the ritual performance (Example 3)
2. Clear interpretations, citations or comments (Example 6)
3. Sentences that are ambiguous with respect to these classes, or sentences that contain elements of both classes (Example 7)

(7) The wife of the chief mourner [...] will carry a symbolic mat that represents the bed of the deceased [...].

We performed an annotation study on a randomly selected ritual description (40 sentences) and found that 15% of the sentences contain both interpretative and factual elements or are ambiguous (clear interpretations: 17.5%, clear factual statements: 67.5%). We did not yet experiment with automatic tagging of sentences according to their class. One possibility, however, could be the application of methods used for the automatic detection of hedges. Academic writers tend to use a high amount of hedges [14]. From the examples in our ritual descriptions, hedges indeed appear quite often. Following the definitions given in [19] and [18], 42.9% of our sentences with a clear interpretative character contain linguistic hedges. There is existing work on the automatic detection of hedges [19, 30] which may be adapted to our specific concerns.

As a first partial solution to the problem, we decided to annotate the clear interpretative sentences as such. They will be ignored for the frame annotation, but remain in the texts.

4 AUTOMATIC LINGUISTIC PROCESSING

As a basis for semantic annotation and processing, the ritual descriptions are preprocessed with standard NLP tools. We use UIMA[5] as a pipeline framework, in which we have integrated a rule-based tokenizer, the OpenNLP[6] part of speech tagger, the Stanford Lemmatizer [31] and the OpenNLP chunker.

4.1 Tokenizing

Many of our texts contain special, non-English characters (*ś*) or complete tokens (*Gaṇeśa*). Therefore, we employ a rule-based tokenizer that uses Unicode character ranges in conjunction with an abbreviation lexicon to detect common abbreviations such as *etc.* or *i.e.*

4.2 Part of speech tagging and chunking

Using standard models for part of speech tagging and chunking produces rather poor results. This is due to the fact that our data contains (i) a lot of unseen tokens and (ii) a high amount of rare and uncommon constructions. We experimented with different scenarios for the domain adaptation of an existing part of speech tagger and chunker.

[5] http://incubator.apache.org/uima/
[6] http://opennlp.sf.net

As we aim at a culture- and source language independent framework, we decided to use a statistical part of speech tagger and chunker, that can be trained on specific corpora.

Large amounts of training material for both labeling tasks are available from other domains, and the annotation of small amounts of data from the ritual domain is feasible. This corresponds to the scenario of fully supervised techniques for domain adaptation discussed in the literature [6]. We experimented with different combination techniques, which are outlined in the following section.

4.2.1 Data sets

Our training data comes from two different sources. We manually annotated 408 sentences of our ritual descriptions with part of speech tags and chunks, using the Penn Treebank tagset and the CoNLL 2000 style of marking chunks [28]. As a second domain corpus we chose the Wall Street Journal, which features compatible part of speech and chunk annotations. For the extraction of chunks from the Penn Treeebank we made use of the CoNLL 2000 scripts. They were also used for the evaluation of the chunker.

We used 10-fold cross-validation to evaluate the data. In order to make sure that our test data did not include any non-ritual data, we "folded" the ritual before mixing it with the Wall Street Journal data. The significance tests are performed against a significance level of $\sigma = 0.95$ using approximate randomization [21, 22].

Table 1. Training sets for part of speech tagger and chunker

Name	Description	Sentences (one fold)	Tok./S.
WSJ	The Wall Street Journal	43,411	27.2
RIT	Ritual Descriptions	343	22.0
WSJ + RIT	Union	43,754	
WSJ + RIT ↑	oversampling RIT	86,822	
WSJ ↓ + RIT	undersampling WSJ	734	
WSJ × RIT	Combined feature space [6]	24,716	
WSJ × RIT ↑	oversampling RIT	50,785	
WSJ ↓ × RIT	undersampling WSJ	702	

Table 1 shows the different data sets and the sizes of one (average) training fold. WSJ + RIT is a simple union of the two sets. As the sizes of the two data sets differ vastly, we also experimented with equally sized corpora, by use of over- and undersampling. WSJ + RIT ↑ represents the union of the WSJ with the oversampled RIT corpus, WSJ ↓ + RIT stands for the union of the undersampled WSJ corpus with the RIT corpus.

The data set WSJ × RIT was produced by augmenting the feature space along the lines of the work in [6]. Let $\vec{v}_i =< f_1, f_2, ..., f_n >$ be the original feature vector for item i and d be a function returning an identifier for a domain. $d(0)$ is then a string representing the general domain, $d(1)$ the domain of rituals and $d(2)$ the domain of news articles. $f_k^{d(x)}$ is the same feature value as f_k, but prefixed with $d(x)$, a domain identifier. The augmented feature vector is then $\vec{v}'_i =< f_1^{d(0)}, f_2^{d(0)}, \ldots, f_n^{d(0)}, f_1^{d(i)}, f_2^{d(i)}, \ldots, f_n^{d(i)} >$, with $i = 1$ or 2. This way, each training example is annotated with a general domain feature vector and a domain specific feature vector. The learner then can learn whether to use the general domain feature set (for which it has massive training data) or the domain specific feature set (with small training data). Again, we used the same over- and undersampling techniques as before.

4.2.2 Evaluation

Part of speech tagging. Table 2 lists the results obtained with training the POS-tagger on different data sets. We use the model trained on the WSJ data set only, i.e., without any domain adaptation, as a baseline. Its performance is 94 % accuracy.

Table 2. Part of speech tagging results with different models

Training data	Accuracy
WSJ	94.01 %
RIT	95.47 %
WSJ + RIT	97.32 %
WSJ + RIT ↑	**97.59 %**
WSJ ↓ + RIT	96.97 %
WSJ × RIT	**97.19 %**
WSJ × RIT ↑	**97.40 %**

If RIT is used as (a small) training set, the POS tagger achieves a performance of 95.47 %. Training on the union of RIT and WSJ yields a significant increase in performance (97.32 %) compared to RIT. Balancing the training sets has minor, but significant influence in both directions.

Augmenting the feature space does not yield significant improvements. Neither the difference between WSJ + RIT and WSJ × RIT nor the difference between the two augmented models is significant.

Table 3. Chunking results with different models

Training data	Precision	Recall	$F_{\beta=1}$
WSJ	87.72 %	87.23 %	87.47
RIT	91.09 %	89.85 %	90.47
WSJ + RIT	90.18 %	89.44 %	89.80
WSJ + RIT ↑	91.07 %	89.62 %	90.33
WSJ ↓ + RIT	91.46 %	90.34 %	**90.89**
WSJ × RIT	88.98 %	88.15 %	88.56
WSJ × RIT ↑	91.75 %	90.24 %	**90.99**
WSJ ↓ × RIT	91.49 %	90.44 %	**90.96**

Chunking. Table 3 shows the results of the chunking models trained on the different data sets. The model trained on both the undersampled Wall Street Journal and ritual descriptions (WSJ ↓ + RIT) performed significantly better than most of the other models (90.89). The two models RIT and WSJ + RIT ↑ perform slightly lower, while not significantly different from each other. The WSJ-model achieves an F-score of only 87.47 and is thus the model with the lowest performance. Using unbalanced data (WSJ + RIT) scores significantly lower than balanced data.

The use of an augmented feature space with balanced data, as represented by data sets WSJ × RIT ↑ and WSJ ↓ × RIT, performs slightly, but not significantly, better than the best standard model. The augmented model used with an unbalanaced data set (WSJ × RIT) performs even lower than the same data set with un-augmented data (88.56).

4.3 Anaphora and coreference resolution

In order to extract continuous and consistent event chains, it is necessary to link anaphoric expressions such as pronouns (8) to their antecedents. In order to study overall performance and potential out-of-domain effects, we applied several anaphora and coreference resolution systems to the same ritual description and evaluated the labeling results.

(8) Let him give a golden coin as ritual fee [. . .].

4.3.1 Candidate systems

GuiTAR [23] and BART [33] are both modular toolkits for experimenting with different algorithms that generate entire coreference chains.

GuiTAR contains an implementation of the rule-based MARS pronoun resolution algorithm [20] and a partial implementation of an algorithm for resolving definite descriptions [34]. The latter part of GuiTAR uses the Charniak parser for preprocessing [5].

BART is a machine learning toolkit which uses a variety of features [29] to train a maximum entropy learner. In order to extract all features, the data need to be parsed or at least chunked. Additional features can be extracted from knowledge resources such as Wikipedia. In our experiment, we did not exploit BART's tuning possibilities but used the standard classifier.

In contrast to BART, JavaRAP is a rule-based anaphora resolution system that implements the Lapping & Leass algorithm for pronominal anaphora resolution [17]. It exclusively treats third person pronouns and lexical anaphors like reflexives and reciprocals and recognizes pleonastic pronouns. While BART and GuiTAR compute full coreference chains, JavaRAP only generates pairs of anaphors and antecedents. JavaRAP also uses the Charniak parser for preprocessing. Here, sentence splitting for parsing was done manually.

4.3.2 Evaluation

We evaluated these systems on a sample ritual description consisting of 295 sentences. We exclusively evaluated the resolution of personal and possessive pronouns in third person such as *it* or *him*. Anaphors which occur in direct speech are disregarded. This leaves us with 18 anaphors for evaluation.

For JavaRAP, we only evaluated anaphora-antecedent pairs. Such a pair was considered correct if the anaphoric relation to (one of potentially several) antecedents was correct. We measure an accuracy of 55.6% correctly resolved pronoun-antecedent pairs. Although this is a reasonably good outcome, the system does not build coreference chains, hence only delivers partial information.

For GuiTAR and BART, we evaluated the coreference chains which contain at least one anaphora using the scorer implemented for the SemEval-2010 coreference resolution task [25]. We measured the standard precision and recall for mention identification, and the MUC precision and recall metric for coreference resolution [35]. The MUC metric emphasizes the correctness of links between mentions within coreference chains while barely penalizing incorrect links between different chains [7]. More specifically, MUC precision is calculated by dividing the number of links in the system output that match the manual annotations by the total number of links in the system output. MUC recall is the ratio between the number of links common to the manual annotation and the system output and the total number of manually annotated links.

As a baseline, we applied the simple heuristic of resolving a pronoun to the nearest preceding noun phrase, without considering further syntactic or morphological information.

Table 4 shows the results of the mention detection task. A mention is identified as strictly correct if the system returns exactly the token of the gold standard. If a system identifies a substring of a gold mention, it is counted as partially correct. The sum of strictly and 0.5 times partially correct identified mentions is used as number of true positives.

As we can see, BART correctly identifies most of the mentions (see the low number of false negatives), however, it tends to overgenerate, with a high number of 'invented' mentions (false positives).

Table 4. Evaluation results for mention identification

Measure	Baseline	GuiTAR	BART
Total found (of 41)	36	52	60
Strictly correct	21	30	35
Partially correct	0	1	1
False positives	15	21	24
False negatives	20	10	5
Precision	58.33%	58.65%	59.16%
Recall	51.21%	74.39%	86.58%
$F_{\beta=1}$	54.54	65.59	70.29

GuiTAR both invents and identifies less mentions than BART. Both systems perfom well above the baseline system.

Table 5. Evaluation results for coreference resolution

Measure		Baseline	GuiTAR	BART
MUC	P	16.66%	50.0%	52.72%
	R	8.82%	61.76%	85.29%
	$F_{\beta=1}$	11.53	55.26	65.16

Table 5 shows precision, recall and f-measures using the MUC metric for coreference resolution. In terms of precision, BART outperforms GuiTAR by 2.72%. Comparing the recall values, BART scores more than 20% higher than GuiTAR.

Error analysis. Investigation of the analyses showed that – across all systems – the proposed coreference chains often contain correct anaphora-antecedent pairs. However, these are extended to incorrect chains, by further linking them to wrong noun phrases and pronouns as antecedents. Example 9 shows a snippet of a coreference chain computed by BART, which resolves an anaphor both correctly and incorrectly.

(9) Continue worship of the ancestors$_1$. Now at the auspicious time bring the girls$_1$ holding their$_1$ hands reciting the mantra.

Such errors also happen when gender agreement is obviously not fulfilled, as shown in example 10.

(10) The father$_2$ should touch the girl$_2$ [...] Let him$_2$ say:

Generally, both GuiTAR and BART tend to overgenerate, proposing in particular coreference chains that do not contain any anaphors. Although the obtained performance measures represent typical rates for state-of-the-art coreference resolution systems, a precision of less than 60% for the generated coreference chains is insufficient for using automatic coreference resolution on a grand scale and using its results as a basis for computing high-quality event chains. In order to obtain a utilizable coreference resolution component, we are planning to experiment with system combination techniques, such as voting and meta learning, using small amounts of annotated domain data.

5 ANNOTATION OF RITUAL DESCRIPTIONS

We use frame semantics [11] as a representation format to encode the ritual sequences such that each separable action mentioned in the ritual corpus is represented by its own frame. The actors that perform the ritual actions as well as objects, times and places mentioned are annotated as frame roles.

In a first phase, we will start with manual annotations, concentrating on developing a suitable frame inventory for the ritual domain. With an established frame inventory and an initial corpus of annotations, we will train a role semantic labeler [9] to explore automatic or semi-automatic annotation.

5.1 Adaptation of existing resources

To guarantee a consistent encoding of ritual frames, the FrameNet lexicon and ontology is used to deliver a base inventory of frames. We try to map the ritual actions to frames that are already defined in FrameNet. For this sake, verbs found in the ritual descriptions are extracted automatically from the chunked ritual descriptions. They are ordered in semantic groups and subsequently searched for in the FrameNet database. This approach has the advantage that we can make use of a well structured inventory of frames.

Coverage. According to a first estimation reported in [26], over 80% of the verbs mentioned in the ritual corpus are contained as lexical units in FrameNet. However, a closer inspection of the ritual data reveals that numerous terms are only identical at a lexical level, but occur in completely different senses. Moreover, a large number of concepts that are important in ritual descriptions are not dealt with in FrameNet. At the current state of annotation, it is difficult to give comprehensive quantitative statements about the coverage of FrameNet on ritual corpora. However, areas not (or only scarcely) covered by FrameNet include, for example, the fields of preparing and serving food.

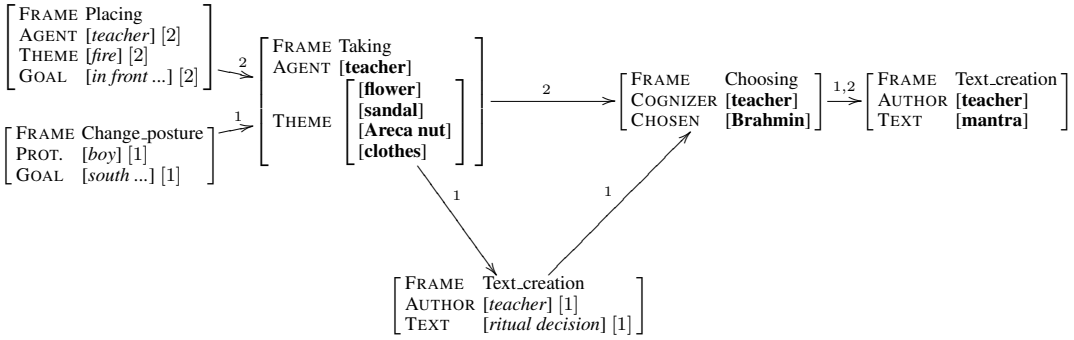
Granularity. Frequently, the frames contained in FrameNet represent concepts that are too abstract for the annotation of rituals. In these cases, FrameNet groups several lexical units into one frame that would not correspond to a single concept in a genuine ritual frame ontology. Examples are the verbs “to cover”, “to anoint” and “to fill”. These verbs are assigned to a single frame *Filling* in FrameNet because they express the same idea of “filling containers and covering areas with some thing, things or substance”. Although we use frames to generalize from the literal level of ritual descriptions, annotating “to fill” and “to anoint” by a single frame in a ritual context would certainly lead to over-generalization and, therefore, to a clear loss of information. New frames have to be designed in such cases. For the example under consideration, we decided to further specify the frame Filling with three more specialized new frames: *Filling_container* (filling a container), *Besmearing_surface* (covering a surface with a liquid) and *Wrapping_object* (wrapping an object with another object).

On the other hand, the granularity of FrameNet frames can also be higher than needed. This case occurs, for instance, in the area of legal concepts, which are covered in great detail by FrameNet. Such cases are easier to resolve than those resulting from coarse-grainedness of frames discussed above, due to the FrameNet inheritance hierarchy. That is, we can use predefined, more abstract frames from higher levels in the hierarchy.

The existence of such selected semantic fields that are covered in FrameNet in great detail clearly demonstrates that it has been successful in modeling specific domains. Thus, for the present project, domain adaptation will consist in modeling finer-grained frame structures for semantic fields that are relevant for the ritual domain.

Annotation and automation. The mapping from verbs to frames is stored in an index and used to assign frames automatically to the

Figure 1. A schematic representation of a common subsequence in two different rituals; the indices indicate the number of the example.



verbs in the preprocessed ritual descriptions. Currently, we have defined 116 of such predefined assignment rules. Applying them to two ritual descriptions yielded coverage rates of 35.2% (479 of 1361 verbal units) for a modern ethnographic report and 82.5% (254 of 308 verbal units) for the translation of an indigenous manual (cf. 3.1), respectively. A closer inspection of the latter reveals that three frames contribute 65.3% to the high coverage. This is caused by the rather monotonous character of this text whose main part consists of the repeated invocation of Hindu deities ("Salutation to god ..."; mapped to a newly designed frame in 88 instances) and describes the recitation of mantras (mapped to FrameNet *Text_creation*, 91 instances) and the offering of ritual stuff to the participants and deities (FrameNet *Giving*, 22 instances).

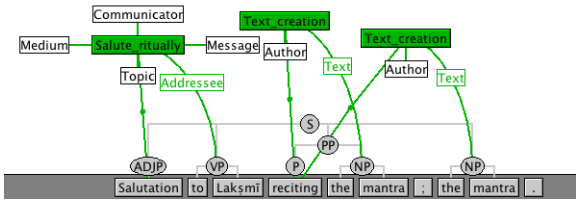


Figure 2. Annotated sentence *Salutation to Lakṣmī reciting the dyāṃ mā lekhīr; the śrīś ca te.*

The Salsa tool [8] (Figure 2) is used to manually correct the automatic annotations and to assign frame-semantic roles to syntactic constituents. This frame semantic information is stored as a separate layer along with the source text and the linguistic annotation. When the corpus text or the linguistic preprocessing layers are updated, this layering mechanism makes it possible to reassign the frame semantic annotation, thus avoiding manual re-annotation.

5.2 Detecting ritual structure

As a proof of concept for the types of analyses we can offer to ritual scientists on the basis of these semantic annotations, we constructed representations for a number of close variations of rituals, two of which are shown below with assigned frames shown as subscripts.

1. "... the boy sits down$_{\text{CHANGE_POSTURE}}$ south of the teacher. (The teacher) takes$_{\text{TAKING}}$ flowers, sandal, Areca nut and clothes and declares$_{\text{TEXT_CREATION}}$ the ritual decision to select$_{\text{CHOOSING}}$ the Brahmin by saying$_{\text{TEXT_CREATION}}$ the mantra ..."
2. " ... (the teacher) places$_{\text{PLACING}}$ (fire in a vessel of bell metal) in front of himself. Having taken$_{\text{TAKING}}$ flowers, sandal, Areca nut, clothing etc. he should select$_{\text{CHOOSING}}$ a Brahmin. The Brahmin is selected with$_{\text{TEXT_CREATION}}$ the mantra ..."

We extracted the event sequences from each description, one starting with PLACING, one with CHANGE_POSTURE. Figure 1 shows a partial semantic representation for the above excerpts. It illustrates one way in which we plan to extract and visualize common subsequences in rituals. The sequences share the frames TAKING, CHOOSING and TEXT_CREATION. Elements occurring in both sequences are printed in bold.

6 FUTURE WORK AND CONCLUSIONS

6.1 Future work

As we have seen, anaphora resolution is currently an unsolved issue. We intend to perform a detailed error analysis of the available systems and to identify strategies and methods that can yield reasonable performance with respect to the overall task.

Several other steps in the preprocessing chain that have not been discussed in this paper need to be addressed in the future. Word sense as well as named entity annotations are needed as a basis for semantic annotation and the structural analysis of rituals. As we established in a pre-study, many ritual specific concepts are not included in sense inventories such as WordNet. Also, named entities occurring in ritual descriptions can often not be classified into the standard classes or do not appear in gazetteer lists. Thus, we expect that both word sense disambiguation and named entity recognition systems and resources need to be adapted to the ritual domain.

Using the types of annotations discussed in this paper, we will create structured and normalized semantic representations for ritual descriptions that are linked to an ontology comprising general-semantic and ritual-specific concepts and relations. This allows us to offer querying functionalities for ritual researchers, so that they can test and validate their hypotheses against a corpus of structurally analyzed ritual descriptions. A well-defined and populated ontology can also be used to automatically identify event sequences in the data.

Sequence analysis and the automatic detection of structure in rituals are the second focus of our future research. As soon as enough data has been encoded in the scheme described in sections 4 and 5, we plan to develop computational methods that support ritual researchers in finding constant patterns and variations in the ritual descriptions. Methods that will be adapted for this purpose include modeling of selectional preferences, as well as algorithms for detecting frequent item sets and statistical tests of significance.

6.2 Conclusions

In this paper, we presented a detailed investigation of the performance of standard NLP tools and resources for the computational linguistic analysis of ritual descriptions. As standard "out of the box" tools perform poorly and lexical resources are lacking coverage and the appropriate granularity, the adaptation of tools and resources to different domains emerges as an important focus of our work. However, we have not only established that standard NLP tools behave poorly on our domain, we also have shown that we can improve the results significantly with rather small effort. This finding supports our basic tenet, that it is possible to make use of computational linguistics methods for the semantic and quantitative analysis of ritual texts. Further work will have to establish whether the representations we compute will allow us to help ritual researchers establish novel insights on the structure(s) of rituals.

Our work also explores to which degree methods of computational linguistics can be adapted to the needs of the Humanities. By using a rarely applied combination of computational and traditional scholarship, we are optimistic to achieve results that extend the knowledge in the field of ritual research to a considerable degree. Moreover, we hope to open up new, more formal data-oriented ways for research in the Humanities.

ACKNOWLEDGEMENTS

This research has been funded by the German Research Foundation (DFG) and is part of the collaborative research center on ritual dynamics (Sonderforschungsbereich SFB-619, Ritualdynamik).

REFERENCES

[1] A. Burchardt, A. Frank, and M. Pinkal, 'Building Text Meaning Representations from Contextually Related Frames – A Case Study', in *Proceedings of IWCS*, (2005).

[2] A. Burchardt, S. Pado, D. Spohr, A. Frank, and U. Heid, 'Constructing integrated corpus and lexicon models for multi-layer annotations in owl dl', *Linguistic Issues in Language Technology*, **1**(1), 1–33, (2008).

[3] N. Chambers and D. Jurafsky, 'Unsupervised learning of narrative event chains', in *Proceedings of ACL: HLT*, pp. 789–797, (2008).

[4] N. Chambers and D. Jurafsky, 'Unsupervised learning of narrative schemas and their participants', 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*, pp. 602–610, (2009).

[5] E. Charniak, 'A Maximum-Entropy-Inspired Parser', in *Proceedings of NAACL*, (2000).

[6] H. Daumé III, 'Frustratingly easy domain adaptation', in *Proceedings of ACL*, pp. 256–263, (2007).

[7] P. Denis, *New learning models for robust reference resolution*, Ph.D. dissertation, Austin, TX, USA, 2007. Adviser-Baldridge, Jason M. and Adviser-Asher, Nicholas M.

[8] K. Erk, A. Kowalski, and S. Padó, 'The SALSA Annotation Tool', in *Proceedings of the Workshop on Prospects and Advances in the Syntax/Semantics Interface*, (2003).

[9] K. Erk and S. Padó, 'Shalmaneser – a Toolchain for Shallow Semantic Parsing', in *Proceedings of LREC*, (2006).

[10] C. Fellbaum, *WordNet: An Electronic Lexical Database*, MIT Press, 1998.

[11] C. J. Fillmore, C. R. Johnson, and M. R. Petruck, 'Background to FrameNet', *International Journal of Lexicography*, **16**(3), 235–250, (2003).

[12] J. R. Finkel and C. D. Manning, 'Hierarchical Bayesian Domain Adaptation', in *Proceedings of HLT-NAACL*, pp. 602–610, (2009).

[13] O. Hellwig, 'A chronometric approach to Indian alchemical literature', *Literary and Linguistic Computing*, **24**(4), 373–383, (2009).

[14] K. Hyland, 'Hedging in academic writing and eap textbooks', *English for Specific Purposes*, **13**(3), 239–256, (1994).

[15] J. Jiang and C. Zhai, 'Instance Weighting for Domain Adaptation in NLP', in *Proceedings of ACL*, pp. 264–271, (2007).

[16] K. Kipper, A. Korhonen, N. Ryant, and M. Palmer, 'A Large-Scale Classification of English Verbs', *Journal of Language Resources and Evaluation*, **42**(1), 21–40, (2008).

[17] S. Lappin and H. J. Leass, 'An algorithm for pronominal anaphora resolution', *Computational Linguistics*, **20**(4), 535–561, (1994).

[18] M. Light, X. Y. Qiu, and P. Srinivasan, 'The Language of Bioscience: Facts, Speculations, and Statements In Between', in *Proceedings of HLT-NAACL Workshop: BioLINK*, eds., L. Hirschman and J. Pustejovsky, pp. 17–24, (2004).

[19] B. Medlock and T. Briscoe, 'Weakly Supervised Learning for Hedge Classification in Scientific Literature', in *Proceedings of ACL*, pp. 992–999, (2007).

[20] R. Mitkov, *Anaphora Resolution*, Longman, 2002.

[21] E. W. Noreen, *Computer-Intensive Methods for Testing Hypotheses*, John Wiley & Sons, 1989.

[22] S. Padó, *User's guide to* `sigf`*: Significance testing by approximate randomisation*, 2006.

[23] M. Poesio and M. A. Kabadjov, 'A general-purpose, off-the-shelf anaphora resolution module: Implementation and preliminary evaluation', in *Proceedings of LREC*, (2004).

[24] S. Pradhan, W. Ward, and J. H. Martin, 'Towards Robust Semantic Role Labeling', *Computational Linguistics, Special Issue on Semantic Role Labeling*, **34**(2), 289–310, (2008).

[25] M. Recasens, T. Martí, M. Taulé, L. Màrquez, and E. Sapena, 'Semeval-2010 task 1: Coreference resolution in multiple languages', in *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future*, pp. 70–75, (2009).

[26] N. Reiter, O. Hellwig, A. Mishra, A. Frank, and J. Burkhardt, 'Using NLP methods for the Analysis of Rituals', in *Proceedings of LREC*, (2010).

[27] J. Ruppenhofer, C. Sporleder, R. Morante, C. Baker, and M. Palmer, 'Semeval-2010 task 10: Linking events and their participants in discourse', in *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions (SEW-2009)*, pp. 106–111, (2009).

[28] E. F. T. K. Sang and S. Buchholz, 'Introduction to the CoNLL-2000 Shared Task: Chunking', in *Proceedings of CoNLL-2000 and LLL-2000*, (2000).

[29] W. M. Soon, D. C. Y. Lim, and H. T. Ng, 'A machine learning approach to coreference resolution of noun phrases', *Computational Linguistics*, **27**(4), 521–544, (December 2001).

[30] G. Szarvas, V. Vincze, R. Farkas, and J. Csirik, 'The BioScope corpus: annotation for negation, uncertainty and their scope in biomedical texts', in *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing*, pp. 38–45, (2008).

[31] K. Toutanova, D. Klein, C. Manning, and Y. Singer, 'Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network', in *Proceedings of HLT-NAACL*, pp. 252–259, (2003).

[32] M. Verhagen and J. Pustejovsky, 'Temporal processing with the TARSQI toolkit', in *Coling 2008: Companion volume: Demonstrations*, pp. 189–192, Manchester, UK, (August 2008). Coling 2008 Organizing Committee.

[33] Y. Versley, S. P. Ponzetto, M. Poesio, V. Eidelman, A. Jern, J. Smith, X. Yang, and A. Moschitti, 'Bart: A modular toolkit for coreference resolution', in *Proceedings of the ACL: HLT Demo Session*, pp. 9–12, (2008).

[34] R. Vieira and M. Poesio, 'An empirically-based system for processing definite descriptions', *Computational Linguistics*, **26**(4), (2000).

[35] M. Vilain, J. Burger, J. Aberdeen, D. Connolly, and L. Hirschman, 'A model-theoretic coreference scoring scheme', in *MUC6 '95: Proceedings of the 6th conference on Message understanding*, pp. 45–52, Morristown, NJ, USA, (1995).

Automatic Pragmatic Text Segmentation of Historical Letters

Iris Hendrickx and **Michel Généreux** and **Rita Marquilhas** [1]

Abstract. In this investigation we aim to reduce the manual workload by automatic processing of the corpus of historical letters for pragmatic research. We focus on two consecutive sub tasks: the first task is automatic text segmentation of the letters in formal/informal parts using a statistical n-gram based technique. As a second task we perform a further semantic labeling of the formal parts of the letters using supervised machine learning. The main stumbling block in our investigation is data sparsity due to the small size of the data set and enlarged by the spelling variation present in the historical letters. We try to address the latter problem with a dictionary look up and edit distance text normalization step. We achieve results of 83.7% micro-averaged F-score for the text segmentation task and 63.4% for the semantic labeling task. Even though these scores are not high enough to completely replace the manual annotation with automatic annotation, our results are promising and demonstrate that an automatic approach based on such small data set is feasible.

1 INTRODUCTION

Research based on historical texts has a high manual workload. In the CARDS project, private Portuguese letters from 16th to 19th century are manually transcribed into a digital format[2]. The aim of the project is to produce an electronic critical edition and historical-linguistic treatment of the text of the letters. The project main linguistic focus is on discourse analysis and historical pragmatics; private letters seem to be the best possible trace left by the linguistic behavior of speakers of the past in their role of social agents. A small part of the collection is annotated with textual and semantic labels to serve as the basis for corpus-based pragmatic studies.

In the study presented here, the aim is to reduce the manual tasks by automatic processing of the corpus of historical letters. We use the small manually labelled sub-set of letters to develop automatic systems in order to label the other part of the collection automatically. We present our first attempt to apply standard statistical and machine learning methods on the labeled data.

We focus on two consecutive sub tasks within the field of historical pragmatics that are important for the research in the CARDS project. The first task is automatic text segmentation of the letters. Letters are a specific genre of text and most of them follow a rather strict division. In general, a letter has an opening part, a formal introduction, the body of the letter that conveys the actual message, and a more formal closing part. To study the language usage in the letters in depth, it is important to discriminate between these formal and informal parts. This type of task is closely related to topic-based text segmentation [20] or paragraph detection [35]. Here we evaluate a method to discover the formal and informal text segments automatically using a statistical method based on n-grams similar to [4].

The second task is to investigate the formal parts in more detail and study their pragmatic relevance. For this purpose, the formal parts of the letters were further annotated with semantic labels from the Lancaster semantic taxonomy [33]. This semantic labeling system has already successfully been applied to historical English texts [2]. We try to predict the semantic labels automatically. As this type of semantic labeling can be viewed as coarse-grained word sense disambiguation, we take a supervised machine learning approach as previous research on WSD has shown that this a successful approach [22, 12].

This investigation has several challenges. First of all, the data set is small, we only have 502 manually annotated historical letters to work with. This is a small set to serve as training material for automatic methods. Secondly, these letters are handwritten in a time period well before spelling standardization so the texts contain many spelling variations. The letters cover a time period of four centuries aggravating the problem of word variation. Furthermore, there are not many preprocessing NLP tools available for this type of historical text, such as a lemmatizer which would have produced useful and valuable information sources for our automatic approach.

Spelling variation is a huge hindrance for automatic processing of historical texts, for example for automatic text retrieval [13, 23] and corpus linguistics [3]. Hence the first step was to normalize the spelling of the texts. We used dictionary lookup and edit distance as is further detailed in Section 3 on experimental methods. In Section 4 we present the automatic text segmentation. Section 5 details the further semantic labeling of the formal parts of the letters. We conclude in Section 6. In the next section we first give more details about the CARDS project and the data set we used in our experiments.

2 CORPUS OF HISTORICAL LETTERS

Historical research builds on a kind of empirical data that is not easy to process automatically. The sources, as the data are traditionally called, lose much information as they were ripped from their original context. In order to re-build those contexts, historians have to double their attention to original object's material features, which can vary immensely. When we think of written documents as historical sources, all this means that Textual Criticism methods[3] become mandatory [5]. As a consequence, the documents have to be read in the original, or in an original-like support, and the more idiosyncratic features are registered, the stronger the historical interpretation can

[1] Centro de Linguística da Universidade de Lisboa, Lisboa, Portugal, email:iris@clul.ul.pt

[2] http://alfclul.clul.ul.pt/CARDS-fly

[3] Textual Criticism is the discipline that combines several methods (paleography, diplomatics, codicology, bibliography, textual recension and editorial emendation) to publish selected texts in the closest possible way to their authors' final intentions.

become. This means to work physically in the archives and to transcribe the documents by hand into a new, digitalized, highly annotated format. It is an expensive and time-consuming process which doesn't have to receive the extra-load of further manual labeling if a proper automatic operation is designed for it.

In the CARDS project, private letters from 16th to 19th century Portugal are being manually transcribed; its aim is to produce an electronic critical edition and historical-linguistic interpretation of the letters. The CARDS sources are unpublished epistolary manuscripts that were kept within criminal law-suits by the Portuguese Inquisition and the Royal Appeal Court as evidence either of guilty parties, or of innocent ones. The Inquisition scanned law-suits are from the 16th, 17th and 18th century, and the Royal Appeal Court law-suits are from the three first decades of the 19th century. In such a textual collection, the discourse of the lower classes is widely represented since the social elites were openly protected in those *Ancien Régime* times, both by the Church and the Royal powers. To complete the social representativeness of this rather 'popular' letters collection, a small sample of the elite's discourse in letters is also being gathered, coming from powerful families' archives. The resulting corpus contains, for now, a sum of 450K words belonging to 1700 letters. The project will stop at 2000 letters (1.850 letters from law-suits, 150 letters from elite family archives), involving more than 1000 identified different participants (plus the anonymous ones). However, for the investigation presented here, we only use a small subpart of the full corpus, 502 letters which have been fully annotated. In Table 1 we list the chronological distribution of the full set of 2000 letters and the sub-part used here as our data set.

Table 1. Distribution of letters in the different centuries in the complete CARDS corpus and the data set used in this study.

century	# total	# data set
16th	50	11
17th	250	165
18th	700	215
19th	1000	111
total	2000	502

The manuscripts are almost always originals and they are transcribed by the quasi-diplomatic method to XML files (this means that it is only allowed to normalize for word boundaries and i/j, u/v variation). Lacunae, difficult deciphering, abbreviations, diacritics and non-orthographic uses are all kept in the transcription by means of the tags which were specifically developed for primary sources editions by TEI (Text Encoding Initiative[4]) and for epistolary data by DALF (Digital Archive of Letters in Flanders[5]).

The information on social backgrounds and situational contexts, being largely recoverable, are also encoded within a database whose entries are anchored within the letters' XML different files. By following Textual Criticism methodologies while keeping in mind social analyses preoccupations, the CARDS project manages to connect such diverse information as manuscripts' physical layout, original writing, authorial emendations, editorial conjectures, information on variants (when copies or comments also survived), information on the context of the letter's composition (an event within a social context), and information on the participants' biographies.

As has been seen in recent literature on historical pragmatics and discourse analysis, letters are extremely valuable sources for the study of the past social uses of language [29, 11]. Nevertheless, being highly subject to genre constraints, not all textual parts within letters can be considered as similar candidates for an idealized representation of face-to-face interaction. Latitude has to be allowed for the presence of strict social rules applying to the epistolary practice only. As Susan Fitzmaurice [16, page 77], puts it:

> Individual writer's letters are as distinctive as their signatures, yet, at the same time, the writers appear to share practices that amount to recognizable conventions of letter writing.

This is true, above all, for opening and closing formulae. Well before approaching the probability of the oral-written parallelism, two questions arise concerning the pragmatic value of these textual conventions: Are they always valid for all social classes, for men as well as for women and for all communicative situations? Or are the conventions culture-dependent and subject to change?

Literary critics of the letter genre seem have strong views about the rigidity of its conventions. Claudio Guillén, when speaking of the "profoundly primitive quality" of the familiar letter in prose, whose origins, in Western culture, go back a great many centuries, to the Mediterranean Antiquity, commented this: "Hence also its highly conventional character, visible since antiquity in the worn and yet indispensable formulas of salutation, apology, recommendation, farewell, and the like, on which both the humble letter writer and the sophisticated poet depend [17, page 81]". Since the CARDS sources are the kind of data that represent the typical "humble letter writer" in a substantial and measurable way, we are in the ideal position to test this kind of assumption. We believe that three hypotheses take form concerning people writing letters in the 16th to 19th century time span: 1. they were just celebrating, over and over, the same rituals when beginning and ending the interaction, regardless of their communicative intention, their social origin, their genre, or 2. they were celebrating the same rituals, regardless of their communicative intention, but social origin and genre were consistent factors of ritual variation, or 3. they chose to celebrate or not the same rituals, depending on their communicative intentions. Current discussion on that pragmatic topic is divided between the *universal* approach, and the *politic* approach. According to the first one, there is a 'pan-cultural interpretability of politeness phenomena' [7, page 283] such as deferential language. But authors taking the *politic* approach choose to disagree: they state that 'deferential language such as terms of address, greetings and leave-taking are weapons in a struggle to 'exercise power' [37, page 156]. The testing of the three hypotheses can become important empirical evidence for the study of ritualized forms of language.

2.1 Annotated data set

In the CARDS project we want to see whether different social agents of the past, wishing to act through their letters (as lovers, as deferential servants, as tyrants, as vindictive judgers, as friends, etc.) used to write or not the same things in the greeting and leave-taking part of their letters. Textual and semantic labeling is then necessary. The textual labeling covers chunks of the CARDS' letter-texts that seem to adopt formulaic beginnings and endings in the light of traditional rhetorics. As for semantic labeling, it is designed to measure the distance between the discursive topics in those formulaic parts on the one hand and on the other hand, to unveil the main topics developed in the letter as a whole textual unit. Accordingly, the CARDS corpus is to be tagged for conventions in the *opener* part of the letters

[4] http://www.tei-c.org/Guidelines/P5/
[5] http://www.kantl.be/ctb/project/dalf/index.htm

(consisting of nomination, date, location, salutations) an formal introduction part (*harengue*), conclusion (*peroration*) and the *closer* part (including signature, nomination, location). In Table 2 we list the number of formal parts in the data set of 502 letters and 177,082 tokens[6].

Table 2. Statistics on the data set of historical letters, frequencies of formal segment types.

segment	frequency
opener	351
harengue	160
peroration	342
closer	434
letters	502
tokens	177,082

These parts are manually annotated with semantic labels. Semantic classification of historical texts can be very biased, so we followed the same choice made by Archer and Culpeper [1] in their study of language usage in a close-to-spoken corpus similar to the CARDS one, namely dialogues in plays and trial proceedings of the 17th and 18th centuries; this means we also applied the UCREL Semantic Analysis System [33]. This taxonomy, abbreviated as USAS, is a conceptually driven schema and has 21 broad categories such as 'body and the individual','time' and 'social actions, states and processes'. Each of these broad categories is subdivided in more fine-grained categories. A sub-set of the CARDS corpus has been manually labelled with a subset of 15 fine-grained labels from the USAS scheme presented in Table 3. Four annotators independently from each other annotated the data according to annotation guidelines and a manual revision was made in the end by one annotator only.

In example (1) we show a sentence from a peroration of a letter written in 1636 by a merchant to his cousin. The author of the letter apologizes to the receiver for bothering him (labeled as politeness S.1.2.4), expresses respect by addressing him with a formal title (S7.2) and wishes that God (labeled as religion S9) will bless him with a long life (a reference to health, labeled as B2)[7].

(1) Comtudo eu <S1.2.4>não deixarei enfadar a Vm</S1.2.4> Com Couzas minhas pois <S7.2> tenho Vm por meu Sro </S7.2><S9>a quem elle gde</S9> Com <B2>larguos annos de vida</B2>

To validate the manual annotations of the USAS labeling, we performed a small experiment. We selected a random sample of 20 letters from the CARDS data set and had three annotators label the tokens in the formal parts independently of each other. Next we computed the inter-annotator agreement on the three different annotations. Although we used just a very small sample, this gave us some indication of the difficulty of the task and the consistency of the labeling of the annotators. We computed the kappa statistics [8], which resulted in a kappa value of .63 for the average on the pairwise comparisons of the three labeled sets. The scores are not very high, but this in line with other semantic labeling tasks such as word sense disambiguation [30].

[6] Punctuation was not counted.

[7] *English translation: However, I won't bother Your Honor with things of mine since I have Your Honor as my Lord whom he (God) guards with many years of life.*

We did observe that there is also some room for improvement of the manual labeling as several disagreements are caused by placing boundaries at different points. Refining the annotation guidelines on what concerns fixed expressions might lead to somewhat higher agreement scores.

Table 3. Sub-set of USAS labels used in our investigation.

Semantic Group	Sub-classification
A: General names	A9: Possession
B: Body	B2: Health and diseases
E: Emotional	E2: Liking
	E4: Happy/sad
I: Money and Commerce	I1: Money
Q: Linguistic	Q1.1.: Communication
	Q2.2.: Speech acts
S: Social	S1.1.1 General (goodbye)
	S1.2.4.:Politeness
	S3.1.:Relationship in general
	S4: Kin
	S7.1: Power, organizing
	S7.2.: Respect
	S9: Religion
X: Psychological	X1: General

3 EXPERIMENTAL SETUP

This section presents our experimental setup, the preprocessing step of text normalization and the resources used in our experiments. To perform our experiments, we randomly divided the manually annotated data set of 502 letters in 100 for testing and 402 for training. Some of the letters in this data set are written by the same person. To prevent unfair overlap between training and testing data, we made sure that none of the test letters was written by the same person as a letter in the training data.

One particular feature of our data is the lack of sentence boundary information. The detection of sentence boundaries is a basic preprocessing step when automatically processing text documents in NLP. Unfortunately, our data has such characteristics that this becomes a difficult task. First of all, the letters do not contain much punctuation marks at all. They date from the time before spelling or writing conventions were widely used, and several of the letters are written by people who hardly had learned how to write. Secondly, most of the letters do not signal the beginning of a sentence with a capital letter. Therefore, we can not use the standard sentence boundary detection techniques that rely on punctuation and capitalization clues such as the work of [34] or [27]. We do split off punctuation marks from words when they are present and otherwise the letters are considered as a long string of words.

Typical for the text genre of letters is a large amount of proper names. Names are more informative as a category than as a word, since the proper noun itself does not tend to reappear across texts. We have made an attempt to tag proper nouns on the basis of an ad hoc list of 6923 Portuguese names selected from diverse sources. Each proper noun was replaced by the tag *PROPERNOUN*.

To normalize the text further, we took the following steps. In the manual annotation, abbreviated words were labeled as such. We extracted a list of these abbreviations from the training set, in total 1224, excluding adverbial abbreviations as these can still have much variance in them (e.g *principalmte, principlmte* for *principalmente*).

The main resource for the text normalization step was a historical Portuguese dictionary [6] of more than 161,000 entries. Rafael Bluteau's Vocabulario is a large Portuguese-Latin encyclopedic dictionary occupying ten in-folio volumes. It was written in the late 17th century and published in the early 18th century by an English born priest. The dictionary importance, today, comes mostly from its author being a careful producer of metalinguistic, comparative judgements on the Portuguese language of the Modern period. For each word in the letters we checked whether it was either an abbreviation or present in the dictionary. Each word that was not, was considered being a spelling variation. These words were replaced with one of the top 5000 most frequent entries in the dictionary having the smallest edit distance, also known as the Levenshtein distance [25]. We limited the maximum edit distance to 5, otherwise the word form was kept.

We performed a small manual validation of the text normalization. We randomly selected 20 letters, 4354 tokens, from the data set and compared the original texts against the normalized versions. On this small sample, the normalization procedure (including proper name detection) achieved an accuracy of 89.7%, recall of 97.5% and a precision of 91.8%. The majority of the precision errors is due to mismatches between the proposed normalized form of a word and the actual correct form, and 9% of these were (capitalized) words mislabeled as names. Two-third of the errors in the recall (false negatives) are due to unrecognized names.

As an additional resource in our experiments, we used the Tycho Brahe Parsed Corpus of Historical Portuguese[8] (TBCHP), an electronic corpus of historical texts written in Portuguese. The corpus contains prose from different text genres and covers a time period from the Middle Ages to the Late Modern era. TBCHP contains 52 works source texts but not all of them are annotated in the same way. Some of the texts maintain original spelling variation, while other texts, intended for part-of-speech and syntactic annotation, were standardized for spelling. Both modern Brazilian and European Portuguese orthographies are represented in this standardization, depending on the chosen printed sources, but the difference between these two writing standards involves only a few cases of vowel accentuation and a few digraphs within classical loan words. From the whole TBCHP corpus of 52 titles, we discarded the texts dated before 1550, given the need to avoid anachronic generalizations. We also discarded most of the texts with non-modernized spelling, thus arriving at a sample that contains 33 texts (12 of which are letter collections) and 1,516,801 tokens.

We also trained a part-of-speech (POS) tagger on the subpart of TBCHP using MBT [9, 10]. As not all texts in our sample are annotated with part-of-speech, we used here a sample of 23 texts (11 of which are letter collections) and 1,137,344 tokens. We only used 45 coarse-grained POS tags and left out specifications of verb tenses or gender/number distinctions. To estimate the performance of the POS tagger, we arbitrarily split the TBCHP data set in a training set of 917,467 tokens and a test set of 219,877 tokens. The measured accuracy on the test set is 95.1%. We cannot estimate the performance of the tagger on the CARDS data as we do not have gold standard labeling of POS tags available here. We expect the accuracy of the POS tagger to be somewhat lower as we switch to another corpus of a specific text genre.

[8] http://www.tycho.iel.unicamp.br/∼tycho/corpus/en/index.html

4 TEXT SEGMENTATION

The segmentation task is to assign each word (1-gram) of the historical letters a single of five classes: four formal segment types, (*opener*, *closer*, *harengue* or *peroration*) and one class for words that do not belong to any formal classes (*free*). Our approach is slightly counter-intuitive, as we rely on lexical models for each class we are trying to identify. There are two compelling reasons for this:

1. Albeit small, we have at our disposal texts that are explicitly marked with segment boundaries, so we are in a position to exploit a fully supervised approach, a position very few researchers were in to tackle the segmentation task.
2. Our lexical models will provide us with a useful source of information to characterize each class.

A more intuitive approach would be to look for lexical patterns allowing the identification of the topic boundaries [20, 15], possibly including a variety of knowledge bases (textual indicators, information linked to syntax or discourse) [35]. Argumentative zoning is another approach [36, 18] making use of lexical cues to identify boundaries. Given the complexities of our corpus, an interesting option is a hybrid approach exploiting *a priori* rules concerning the beginning and ending of the formal parts of the letters. Our approach, similar to [4], is to assign each n-gram (n $\leq$ 3) in the training data a score representing its salience for the class in which it appears. These scores are used to compute the best class for each word. We use the log odds ratio as a statistical measure of salience or prominence of a n-gram. The log odds ratio measure [14] compares the frequency of occurrence of each n-gram in a given specialized corpus with its frequency of occurrence in a reference corpus as given in equation (2) where a is the frequency of a word in the specialized corpus, b is the size of the specialized corpus minus a, c is the frequency of the word in the general corpus and d is the size of the general corpus minus c.

$$ln(ad/cb) = ln(a) + ln(d) - ln(c) - ln(b) \quad (2)$$

High positive log odds scores indicate strong salience, while high negative log odds scores indicate words irrelevant for the class. We constructed five specialized corpora, one for each of the five classes: *opener*, *harengue*, *peroration*, *closer* and *free*. We adopted the TBCHP (described in Section 3) as our reference corpus. The TBCHP is a good reference corpus for three main reasons:

1. It is quite diversified in terms of genres while CARDS only has private letters.
2. It is almost entirely populated by samples of formal literary Portuguese; on the contrary, CARDS is quite varied in terms of registers (formal and informal) and social representativeness.
3. It is largely standardized for orthography.

The training set was used to create the specialized corpora for each text segment class by concatenating, for each letter, all words belonging to one particular class. This produces the specialized training corpora, whose number of texts, relative n-gram distribution and average frequencies (per text) are detailed in Table 4. Not every letter contains all formal parts but most of them contain at least one formal element. There are 45 letters consisting only of free text.

The salience for each n-gram was then computed and sorted from the highest to the lowest. In table 5 we show for each class the most salient 3-grams and its log odds ratio.

Table 4. Statistics on n-grams in the specialized corpora for the 5 segment classes. $\overline{fq}$ = average frequency of the n-grams.

corpus	# texts	1-gram ($\overline{fq}$)	2-gram ($\overline{fq}$)	3-gram ($\overline{fq}$)
opener	275	1.1% (3.1)	0.9% (1.5)	0.7% (1.3)
harengue	121	2.3% (3.6)	2.2% (1.3)	2.2% (1.1)
peroration	231	2.6% (4.0)	2.5% (1.4)	2.4% (1.1)
closer	343	3.3% (3.9)	3.1% (1.6)	2.9% (1.2)
free	402	90.7% (11.9)	91.3% (1.7)	91.8% (1.1)
total	1372	100%	100%	100%

Table 5. Most salient 3-grams for each of the four formal segment types.

Segment	3-gram	
opener	*Illmo e Exmo*	'most illustrious and excellent' 12.9
harengue	*da q me*	'of the (fem) that I' 8.6
peroration	*Ds gde a*	'God guards to'11.6
closer	*De V Exa*	'of your excellency' 10.8

4.1 Classifying each word

The lists of n-grams with salience values for each class constitute our language models for our classifiers. To derive one particular class tag for each word, the classifier adopts the following two-step strategy:

1. Each 1-gram is assigned salience values for each class as found in the model, zero otherwise;
2. Each word of a n-gram (n $\in \{2,3\}$) has its salience values augmented by the corresponding salience values for the n-gram in the model, if they exist.

The above procedure can be restricted to a subset of one (1-grams, 2-grams or 3-grams only), two (1 and 2-grams only, 1 and 3-grams only or 2 and 3-grams only) or three (1, 2 and 3-grams) models. Therefore, each word from the letter has a salience value for each class, possibly taking into account contextual information (if models above 1-gram were included in the computation process). At this point, one could simply select the class with the highest salience value, but we decided to include a further step in the computation in order to give a fairer evaluation of the salience. The highest value was decreased by the value of salience from other classes. For example, if a word has the following salience values: opener:2, closer:5, harengue:-4, peroration:1 and free:0, it is classified as a *closer* (the largest value) with:

$$salience = 5 - 2 - (-4) - 1 - 0) = 6$$

We evaluate the performance of our classifier on the test set of 100 letters. We present F-scores and overall accuracy computed at the word level. We also computed the micro-average F-score over the five classes. Results are shown in Table 6. The 3-gram model clearly achieves the best results.

4.2 Segment production

The previous approach to classify words from a letter on an individual basis can produce rather sketchy classification in which *holes* exist that could otherwise be used as connectors between distant members of the same class. Let's look at the following hypothetical example where subscripts[9] indicate each word tag (as computed by the classifier) and bracketed tags indicate the true tag (as annotated by humans)[10]:

<opener> Meo$_o$ amo$_o$ e$_c$ Sr$_o$ </opener> <harengue>Ainda$_c$ q$_h$ VM$_f$ me$_h$ não$_h$ quer$_h$ dar$_h$ o$_c$ alivio$_h$ de$_h$ suas$_h$ novas$_h$ a$_h$ minha$_h$ amizade$_h$ não$_h$ pide$_h$ tal$_h$ discuido$_c$ e$_h$ assi$_h$ lembresse$_h$ VM$_h$ de$_o$ mim$_f$ q$_h$ com$_h$ novas$_h$ suas$_h$ q$_h$ bem$_h$ sabe$_h$ q$_f$ não$_h$ tem$_h$ qm$_p$ lhas$_h$ dezeje$_h$ com$_h$ mais$_h$ veras$_p$. </harengue> Sabado$_f$ nove$_f$ deste$_f$ mes$_f$ Domingos$_f$ ... por$_f$ não$_f$ ficar$_f$ com$_f$ escrupello$_f$ <peroration> aqui$_p$ fico$_p$ ás$_h$ ordens$_p$ de$_p$ VM$_p$ pa$_p$ o$_f$ q$_p$ me$_p$ quizer$_p$ mandar$_p$ com$_f$ gde$_p$ vontade$_p$ Ds$_p$ gde$_p$ a$_p$ VM$_p$ </peroration> <closer> Prada$_f$ 10$_f$ de$_f$ Julho$_c$ de$_c$ 1712$_f$ Mayor$_f$ Amo$_c$ e$_c$ Servidor$_c$ de$_f$ VM$_c$ Frando$_f$ de$_c$ SáMzes$_f$ </closer>

Although the patterns of computed tags follow roughly the true annotation, a smoothing technique could be applied to attempt to fill the gaps and create boundaries approaching those created by human annotators. We now explain the approach adopted for smoothing the patterns, inspired from techniques used in sequence modeling [24, 26]. First, we need to have an idea of the average word length and standard deviation of the formal text segments (in words). We obtained those statistics from the training data, shown in table 7.

Table 6. F-scores and overall accuracy for text segmentation.

n-gram used	F-scores% op	har	per	clo	free	Microav.	Overall acc
{1}	3.4	13.2	5.5	22.3	66.2	60.1	50.2
{2}	8.0	17.5	7.5	30.7	71.5	65.4	56.7
{3}	31.1	17.2	13.7	25.6	91.8	83.7	87.0
{12}	4.5	12.1	7.7	23.2	56.9	52.0	41.0
{13}	4.2	17.3	8.1	27.4	64.8	59.3	49.2
{23}	8.3	19.8	8.0	32.8	72.0	66.0	57.5
{123}	4.7	13.1	8.5	24.1	57.0	52.1	41.4

Table 7. Distribution of segment lengths of each class.

Class	Mean	St. Deviation
opener	5.28	4.5
harengue	25.18	17.48
peroration	15.35	14.33
closer	13.69	6.91

[9] o=opener, c=closer, h=harengue, p=peroration and f=free

[10] English translation: <**opener**> My friend and Lord </**opener**> <**harengue**> Although Your Honor does not want to relieve me with news from Your Honor, my friendship does not call for such a lack of attention, so, remember to give me some news, because you know well that there is nobody else that desires them more than I do, really </**harengue**> Saturday 9 of this month ... of not having scruples <**peroration**> here I remain at the orders of Your Honor for all that Your Honor chooses to command, with all my good will, God keeps Your Honor </**peroration**> <**closer**>Prada, 10th of July 1712 The greatest friend and servant of Your Honor Fernando de Sá Menezes </**closer**>

The combined values for mean and standard deviation will give us an idea of the size of the segments for each class we should be aiming at, on average. We choose an interval for the length of each class so that 95% of the segment size falls within the interval. This is given in normal distribution by computing (mean $\mp$ 2$\times$standard deviation). This formula does not hold when distribution are skewed, but we will neglect skewness for our purpose. Computing intervals for each class, we have: [1,15] for *opener*, [1,28] for *closer*, [1,60] for *harengue* and [1,43] for *peroration*. This means that we will consider only *openers* of size ranging from 1 to 15 words, etc.

Starting from the first word of each historical letter, we compute a score for each segment of each of the four classes, considering the length intervals defined previously for each class. A score and a class for each segment are obtained by keeping the class for which the sum *S* of the score of each word within the segment is the highest. Put simply, a segment of *N* consecutive words is likely to be labeled with class *C* if it has many words with high salience values for *C*. We keep the segments above a certain threshold for *S* that do not overlap, otherwise the segment is deemed to be *free* text. We know from observation in the training set that there should be no more than one *opener*, one *closer*, maximally two *harengues* and maximally two *perorations*, and the sequence ordering must be *opener*, *harengue*, *peroration* and *closer*. We evaluated the performance of this smoothed classifier (also at word level) on the same 100 letters. As can be observed in table 8, the scores in general are lower than the results without smoothing.

Table 8. F-scores and overall accuracy for text segmentation after smoothing.

n-gram	F-scores%						Overall
used	op	har	per	clo	free	micro-av	acc
{1}	1.6	2.2	3.4	40.2	52.6	48.3	36.9
{2}	6.5	11.8	6.5	36.8	49.3	45.7	35.1
{3}	33.2	29.5	12.1	32.9	86.0	79.2	77.6
{12}	3.3	5.2	5.1	35.5	62.2	56.8	46.8
{13}	2.4	9.9	7.4	48.8	58.5	54.2	43.3
{23}	7.4	22.2	7.4	42.0	50.0	46.9	35.9
{123}	3.5	10.3	7.4	36.8	63.4	58.1	48.2

Although accuracies are clearly above what could be expected from a random baseline (five classes $\rightarrow$ 20%), at the same time they are clearly below a majority baseline (*free* $\rightarrow$ 91%) or an "average" baseline (89%)[11]. The figures for F-scores from tables 6 and 8 with regards to the four classes of interest are somewhat disappointing but not surprising: we only have a small data set to work with. Nevertheless, we think that our approach is a good starting point. Results from tables 6 and 8 also suggest the following interesting observations:

- In general, larger n-grams can make a better discrimination between the five classes.
- *Free* and *closer* are the classes which can be discriminated best, while *opener* and *peroration* are most difficult.
- Smoothing does degrade the overall word classification performance, although it notably improves F-scores for the *closer* class in general, and, when focusing on the trigram results also for *opener* and *harengue*.

[11] The average baseline is obtained by segmenting the text in the canonical order: *opener*, *harengue*, *free*, *peroration* and *closer*. Then the size of each segment is computed in accordance with the average size of each segment in a text: 1%, 8%, 84%, 4% and 4% respectively.

Clearly, an evaluation of the approach at the segment level would give a clearer picture of the results. However, to our knowledge there is no evaluation metric for segmentation using multiple classes, so future work should look at how existing metrics such as WindowDiff [31] or those proposed in [15] could be adapted for this task.

5 SEMANTIC TAGGING

For the semantic labeling we focused on the formulaic parts of the letters. The semantic labeling of words in the letters is related to coarse-grained word sense disambiguation. Within the formal parts, approximately 56% of the words in the training set has been labeled with a semantic label, the other words not belonging to the 15 semantic classes listed in table 3 are labeled as 'O'. Only 515 words in the training set are ambiguous and occur with more than one possible label. We zoomed in on these ambiguous words as they are the most interesting ones. The average frequency of these ambiguous words in the training set is 17.3 and on average they have 3.3 different labels. Only 155 of the 515 words occur at least 10 times, this low number again confirms the small size of the data set we are working with.

This semantic labeling task also has ties to multi-word expression recognition. Each word is assigned a semantic tag, but often these words are part of situational fixed expressions as the formal part of the letters consist mostly of ritual utterances [28]. The average length of the labelled semantic expressions is 3.4 tokens (measured on the training set).

In a general supervised WSD approach, contextual features are the most important information source [21]. In such a standard WSD approach, the words left and right from the target words, their part-of-speech and/or lemma information are represented as a feature vector for training a machine learning algorithm. We adopted this strategy and used local context information of neighboring words as our main information source in a ML approach. We used the POS tagger described in Section 3 to predict POS tags. We created a feature vector for each word in the text consisting of three words and predicted POS tags left and right of the target word. As our data is sparse, we tried to grasp the general orthographic properties of the target words in set of standard orthographic binary features: starts-with-capital, all-capitals, all-lower-case, contains-number. We also used word length and prefix and suffix information (of 2 and 3 characters) as features.

We tested whether we could use the prefix information as a rough form of root/lemma information source. For each target word and its context words we added a prefix of three characters as additional feature.

We also have for each word the information available whether it occurs in a *opener, harengue, peroration* or *closer* segment. To investigate whether this is a useful feature or not for the semantic labeling task we used the manually annotated segment information in our experiments (the errors in the predicted tags could cloud the results). So for each word and for each word in its 3-1-3 context, we added an additional feature specifying whether it was located in one of the 4 formal segment classes.

We evaluated the different features in the following way. We ran 10-fold cross validation experiments on the training set with three different classifiers and four feature set combinations and measured micro-average F-scores. We chose the following classifiers: Naive Bayes (NB), Decision trees (J48) and Instance-based learning with a search scope of 3 inverse-distance-weighted nearest neighbors (IB3) as implemented in the Weka toolkit [19]. We tried the following feature combinations: the target word and POS and local context (*313*), the addition of the orthographic and suffix/prefix information

of the target word (*313.orth*), the addition of prefixes for all local context words (*313.orth.prefix*), and the combination of local context, orthographic information and the formal segment class features (*313.orth.seg*). The results of the experiments are presented in Table 9. Varying the feature sets has different effects for the different classifiers. Adding the orthographic features improves performance for two classifiers, but doesn't seem to effect J48 much. Adding the prefix features reduces the performance of two of the three classifiers which indicates that using a simple prefix as replacement for lemma information is not sufficient nor helpful. The segment class features, on the contrary, improve the scores for two of the classifiers but not for Naive Bayes.

Table 9. Average F-scores of 10 fold cross validation experiments on the training set with Naive Bayes(NB), Decision trees(J48) and instance-based learning with k=3 (IB3) with four different feature set variants.

feature set	NB	J48	IB3
313	62.2	64.5	61.5
313.orth	65.7	64.5	67.7
13.orth.prefix	65.2	64.7	65.0
313.orth.seg	64.2	64.8	68.5

Table 10. Baseline F-scores on test set. The last column is the label frequency in the test set.

label	Precision	Recall	Fscore	freq
O	67.7	85.3	75.5	1686
A9	25.0	100.0	40.0	1
B2	54.8	36.2	43.6	174
E2	33.3	8.8	13.9	57
E4	40.0	7.8	13.1	102
I1	0.0	0.0	0.0	1
Q1.1	51.9	28.3	36.6	145
Q.2.2	33.3	11.1	16.7	9
S1.1.1	70.8	39.5	50.7	43
S1.2.4	30.6	38.0	33.9	50
S3.1	54.9	37.3	44.4	75
S4	72.0	31.0	43.4	174
S7.1	59.0	46.0	51.7	50
S7.2	48.3	49.4	48.8	393
S9	56.7	52.4	54.4	479
X1	46.4	40.2	43.0	127
total	61.6	59.9	59.1	3566

For the experiments on the test set we computed the following sharp baseline: we assigned each word in the test set its most frequent label as it occurred in the training set. Words not occurring in the training set were labeled as 'O'. The detailed baseline results for each class label are shown in Table 10. For the most frequent class 'O' the baseline recall is very high as can be expected. Most of the other classes receive an F-score between 35% and 50% and the micro-averaged F-score over all 16 labels is 59.1%.

We evaluated the best setting found on the training set on the test set: (*313.orth.seg*), with instance-based learning. The detailed precision, recall and F-scores for each of the semantic classes separately are shown in Table 11. On the test set, the scores are slightly lower, an accuracy of 63.8% and a micro-averaged F-score of 63.4%. When comparing these results against the baseline scores in Table 10, the machine learning approach has in general higher scores than the baseline. On the most frequent class 'O' the baseline achieved a higher recall, but the learner reaches a better precision leading to an overall higher F-score. For high frequent classes such as *S7.2* and *S9* we expected higher scores than for the lower frequent classes. Indeed for *S9* we found a F-score that is higher than the scores for the other semantic labels. However, the class *S7.2* had a low precision and is wrongly predicted as *O* 150 times and 185 times as one of the other semantic classes. Some semantic classes such as *I1* or *Q2.2* hardly occur at all in our data set, and the scores are consequently zero. For low frequent classes we can expect low scores due to data sparsity. The class *E4* has a remarkable low score considering that it occurs quite frequently. When we look at the confusion matrix, we can observe that this class is confused with almost all other classes; many other class examples have been wrongly predicted as *E4*, and also the other way around. Also *E2* has a low F-score. Also the baseline scores for these two classes are low, around 13% F-score. As both of these belong to the same broad semantic category 'Emotion', our results suggest that class is particularly difficult to classify.

Table 11. F-scores on test set with IB3 for the USAS classes. The last column is the label frequency in the test set.

label	Precision	Recall	Fscore	freq
O	77.2	77.4	77.3	1686
A9	0	0	0	1
B2	54.3	58.6	56.4	174
E2	38.5	17.9	24.4	57
E4	18.2	9.8	12.7	102
I1	0	0	0	1
Q1.1	48.4	53.1	50.7	145
Q2.2	0	0	0	9
S1.1.1	54.2	30.2	38.8	43
S1.2.4	44.2	68	53.5	50
S3.1	47.4	36.0	40.9	75
S4	56.8	43.1	49.0	174
S7.1	63.9	46.0	53.5	50
S7.2	42.9	64.1	51.4	393
S9	69.2	61.8	65.3	479
X1	52.5	40.9	46.0	127
total	64.1	63.8	63.4	3566

6 CONCLUSION

We presented our first results on automatic text segmentation for historical letters. We produced an automatic text segmenter that distinguishes between informal free text and four formal parts of the letters. We also created a classifier that can predict semantic labels for words and fixed expressions in the formal parts. We tried to overcome the main stumbling block of the data set: data sparsity due to the small size of the data set and enlarged by the spelling variation present in the data. We applied a dictionary-based text normalization and replaced names with a placeholder. We achieved 83.7% micro av. F-score on text segmentation and 63.4% on the semantic labeling task. Our current results are not good enough that it can replace the need for manual annotation completely, but they are a promising result.

It is worth mentioning that the n-gram extraction as described in Section 4, combined with semantic labeling information, already gave us a valuable resource for further pragmatic studies. An analysis of the most salient n-grams allows us to make the following general comments on the pragmatic function of the formal parts:

- *opener*: salience of the semantics of social respect expressed by nominal addressing forms of deference (for example *Your Excellency*)

- *harengue*: salience of the semantics of health, combined with psychological verbs and phatic expressions, also typical of wishes of good health in the beginning of spoken dialogues (for example *I hope you are in good health*)
- *peroration*: salience of the semantics of religion, combined with phatic expressions, also typical of the God-invoking behaviour in the ending of spoken dialogues (for example *May God be with you*)
- *closer*: once again, salience of the semantics of social respect, expressed here by adjectival and nominal forms of auto-derision (for example *I am your humble servant*).

There is much room for improvement. We applied a text normalization step to reduce word variation. A next step will be to develop a lemmatizer or stemming algorithm that is suited for historical Portuguese text. As Portuguese is a highly inflectional language, lemma information can reduce the data sparsity. For the text segmentation task we would further investigate alternative smoothing techniques and more advanced methods to compute the final segmentation, including lexical models of the shifts between segments. For the prediction of semantic labels we adopted a general word sense disambiguation approach of using local context information and orthographic information. In future work we would like to exploit a more sequential oriented approach that is commonly used for more syntactic tasks such as part-of- speech tagging or chunking [32] . We also plan to work with algorithms designed for sequence modeling such as conditional random fields.

ACKNOWLEDGEMENTS

We would like to thank Mariana Gomes, Ana Rita Guilherme and Leonor Tavares for the manual annotation. We are grateful to João Paulo Silvestre for sharing his electronic version of the Bluteau Dictionary and frequency counts. This work is funded by the Portuguese Science Foundation, FCT (Fundação para a Ciência e a Tecnologia).

REFERENCES

[1] D. Archer and J. Culpeper, 'Identifying key sociophilological usage in plays and trial proceedings): An empirical approach via corpus annotation', *Journal of Historical Pragmatics*, **10**(2), 286–309, (2009).
[2] D. Archer, T. McEnery, P. Rayson, and A. Hardie, 'Developing an automated semantic analysis system for early modern english', in *Proceedings of the Corpus Linguistics 2003 conference*, pp. 22 – 31, (2003).
[3] A. Baron and P. Rayson, 'VARD2: A tool for dealing with spelling variation in historical corpora.', in *Proceedings of the Postgraduate Conference in Corpus Linguistics*, (2008).
[4] M. Baroni and S. Bernardini, 'Bootcat: Bootstrapping corpora and terms from the web', in *Proceedings of Language Resources and Evaluation (LREC) 2004*, pp. 1313–1316, (2004).
[5] A. Blecua, *Manual de Crítica Textual*, Castalia, Madrid, 1983.
[6] R. Bluteau. Vocabulario portuguez, e latino [followed by] supplemento ao vocabulario portuguez. vols. 1-8, I-II. Coimbra-Lisboa. (1712–1728).
[7] Penelope Brown and Stephen C. Levinson, *Politeness: some universals in language usage*, Cambridge University Press, Cambridge, 1987.
[8] J. Cohen, 'A coefficient of agreement for nominal scales', *Education and Psychological Measuremen*, **20**, 37–46, (1960).
[9] W. Daelemans and A.Van den Bosch, *Memory-Based Language Processing*, Cambridge University Press, Cambridge, UK, 2005.
[10] W. Daelemans, J. Zavrel, A. Van den Bosch, and K. Van der Sloot, 'Mbt: Memory-based tagger, version 3.1, reference guide.', Technical report, ILK Technical Report Series 07-08, (2007).
[11] *Studies in Late Modern English Correspondence*, eds., Marina Dossena and Ingrid Tieken-Boon van Ostade, Peter Lang, Bern, 2008.
[12] P. Edmonds and A. Kilgarriff, 'Introduction to the special issue on evaluating word sense disambiguation systems', *Natural Language Engineerin*, **8**(4), 279–291, (2002).
[13] A. Ernst-Gerlach and N. Fuhr, 'Retrieval in text collections with historic spelling using linguistic and spelling variants', in *Proceedings of the ACM/IEEE-CS conference on Digital libraries*, pp. 333–341, (2007).
[14] B. Everitt, *The Analysis of Contingency Tables*, Chapman and Hall, 2nd edn., 1992.
[15] O. Ferret, 'Segmenter et structurer thématiquement des textes par l'utilisation conjointe de collocations et de la récurrence lexicale', in *TALN 2002*, Nancy, (24-27 juin 2002).
[16] S. M. Fitzmaurice, 'Epistolary identity: convention and idiosyncrasy in late modern english letters', in *Studies in Late Modern English Correspondence*, eds., Marina Dossena and Ingrid Tieken-Boon van Ostade, 77–112, Peter Lang, Bern, (2008).
[17] C. Guillén, *Renaissance Genres: Essays on Theory, History and Interpretation*, chapter Notes towards the study of the Renaissance letter, 70–101, Harvard University Press, 1986.
[18] B. Hachey and C. Grover, 'Extractive summarisation of legal texts', *Artificial Intelligence and Law: Special Issue on E-government*, **14**, 305–345, (2007).
[19] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, 'The weka data mining software: An update', *SIGKDD Explorations*, **11**(1), (2009).
[20] M. A. Hearst, 'Texttiling: Segmenting text into multi-paragraph subtopic passages', *Computational Linguistics*, **23**(1), 33–64, (1997).
[21] D. Jurafsky and J. H. Martin, *Speech and Language Processing. 2nd edition.*, Prentice-Hall, 2009.
[22] A. Kilgarriff and M. Palmer, 'Introduction to the special issue on senseval', *Computers in the Humanities*, **34**(1-2), 1–13., (2000).
[23] M. Koolen, F. Adriaans, J. Kamps, and M. de Rijke, 'A cross-language approach to historic document retrieval', in *Advances in Information Retrieval: 28th European Conference on IR Research (ECIR 2006)*, volume 3936 of *LNCS*, pp. 407–419. Springer Verlag, Heidelberg, (2006).
[24] J. Lafferty, A. McCallum, and F. Pereira, 'Conditional random fields: Probabilistic models for segmenting and labeling sequence data', in *Proc. 18th International Conf. on Machine Learning*, pp. 282–289. Morgan Kaufmann, San Francisco, CA, (2001).
[25] V. Levenshtein, 'Binary codes capable of correcting deletions, insertions, and reversals', *Sovjet Physics Doklady*, **10**, 707–710, (1966).
[26] Stephen Merity, Tara Murphy, and James R. Curran, 'Accurate argumentative zoning with maximum entropy models', in *NLPIR4DL '09: Proceedings of the 2009 Workshop on Text and Citation Analysis for Scholarly Digital Libraries*, pp. 19–26, Morristown, NJ, USA, (2009). Association for Computational Linguistics.
[27] A. Mikheev, 'Periods, capitalized words, etc.', *Computational Linguistics*, **28**, 289–318, (1999).
[28] R. Moon, *Fixed Expressions and Idioms in English: A Corpus-Based Approach.*, Oxford University Press, Oxford, 1998.
[29] *Letter Writing*, eds., Terttu Nevalainen and Sanna-Kaisa Tanskanen, John Benjamins Publishing Company, Amsterdam/Philadelphia, 2007.
[30] H. T. Ng, C. Y. Lim, and S. K. Foo, 'A case study on inter-annotator agreement for word sense disambiguation', in *Proceedings of the SIGLEX Workshop On Standardizing Lexical Resources*, (1999).
[31] L. Pevzner and M. A. Hearst, 'A critique and improvement of an evaluation metric for text segmentation', *Comp. Linguistics*, **28**, 1–19, (2002).
[32] L.A. Ramshaw and M.P. Marcus, 'Text chunking using transformation-based learning', in *Proceedings of the Third Workshop on Very Large Corpora*, pp. 82–94, (1995).
[33] P. Rayson, D. Archer, S. L. Piao, and T. McEnery, 'The UCREL semantic analysis system', in *Beyond Named Entity Recognition Semantic labelling for NLP tasks (LREC 2004)*, pp. 7–12, (2004).
[34] J. C. Reynar and A. Ratnaparkhi, 'A maximum entropy approach to identifying sentence boundaries', in *Proceedings of the Fifth Conference on Applied Natural Language Processing*, pp. 16–19, (1997).
[35] C. Sporleder and M. Lapata, 'Broad coverage paragraph segmentation across languages and domains', *ACM Transactions on Speech and Language Processing*, **3**(2), 1–35, (2006).
[36] S. Teufel and M. Moens, 'What's yours and what's mine: Determining intellectual attribution in scientific text', in *In EMNLP-VLC*, (2000).
[37] R. Watts, *Politeness*, Cambridge University Press, Cambridge, 2003.

Semi-automatic Normalization of Old Hungarian Codices

Csaba Oravecz and **Bálint Sass** and **Eszter Simon**[1]

Abstract. An annotated corpus of Old Hungarian is being developed, which requires a number of standard computational language processing tasks: sentence segmentation and tokenization, normalization of tokens and morphological analysis, and automatic morphosyntactic disambiguation. The paper presents how the normalization process of historical texts can be aided with the application of a neat probabilistic model, which renders the output of automatic normalization as a (well constrained) set of legitimate transliterations for each Old Hungarian token, from which a human annotator can select the context fitting element.

1 INTRODUCTION

The availability of annotated language resources is becoming an increasingly important factor in more and more domains of linguistic research: even outside the realms of computational linguistics high-quality linguistic databases can provide fruitful ground for theoretical investigations. Historical corpora represent a rich source of data and phenomena from this perspective but only if relevant information is specified in a computationally interpretable and retrievable way. A major aim of the project which initiated the present research is to produce such an annotated corpus for specific stages of the history of the Hungarian language, similar in purpose but more ambitious in certain respects that the initiatives of eg. [1]. The corpus will contain all the sources from the Old Hungarian period, and carefully selected proportional samples of the Middle Hungarian period, representing various dialects, genres and registers. The project aims to collect and process only the continuous sources, so sporadically found Hungarian words in foreign texts are not considered. The final database will contain 28 smaller texts and 47 codices from the Old Hungarian period.

Work in the first phase starts with the acquisition of source data, part of which has already been converted into some electronic format. However, a significant part is only available in print, where digitization is necessary in the form of mainly manual scanning followed by a conversion process from the scanned images into regular text files aided by OCR software. This step must be completed by extensive manual proofreading and correction to ensure good quality initial resources as input to further computational processing.

For historical corpora, the development of an annotation, which is more or less prototypical in modern language corpora (with sentence boundaries marked and each token supplied with morphological information), requires a number of standard computational language processing tasks: i) sentence segmentation and tokenization; ii) standardization/normalization/regularization of tokens iii) morphological analysis and automatic morphosyntactic disambiguation.

[1] Research Institute for Linguistics, Hungarian Academy of Sciences, Budapest, email: {oravecz,sass.balint,simon.eszter}@nytud.hu

The second step, as the primary focus of the present paper, is inevitable and obviously of critical importance: without normalization the performance of automatic annotation in later stages will suffer a dramatic decrease [2].

1.1 Text normalization

The normalization step has two principal criteria: uniformity and adherence to the original text — at least at the level of morphosyntactic representation. The normalized wordform will be input to the morphological analysis, therefore orthographic variants of the same lexical items in the original text must be neutralized and converted into modern Hungarian spelling as much as possible. This way technology developed for the morphological analysis of Modern Hungarian can be easily applied to the historical texts.

Apart from the fact that normalization is a time-consuming, highly skilled and delicate work, the fact that the style of orthography in old texts and the set of special characters in them are different causes even more difficulties. The Hungarian writing system evolved from the need to translate Latin religious texts, but the adaptation of the Latin alphabet to Hungarian posed several problems. The main one was that there are Hungarian phonemes which do not exist in Latin, so new characters were needed to represent them. The orthography in the 14-16th century was far from uniform, in addition, one codex could have been written by more authors, which causes even more heterogeneity in the texts. The one sound-one letter mapping was rare in the various orthographies, so such a consistency should not be expected from an evolving writing system. Typically, sound-letter correspondences vary a lot even within a single text sample (e.g. *Vylag uilaga* [világ világa]), one letter could stand for multiple sounds (e.g. *zerzete zerent* [szerzete szerint]). In addition, some letters can refer to vowels and consonants as well (e.g. the letters *u,v,w* were used to denote the *u,ú,ü,ű,v* sounds for centuries).

This makes it difficult if not unfeasible to tackle the problem with simple letter replacement rules and underlines the ability of any trained model for generalization to the various language samples. For these reasons, some probabilistic learning paradigms offer themselves naturally, of these Shannon's noisy channel model [3] is one of the cleanest and neatest possibilities, which is easy to formulate and adapt to the normalization task.

For reasons to be detailed later, the normalization process cannot be made fully automatic and will always require manual correction. Thus, the end-to-end application is in effect a preprocessing toolset very similar to the one presented in [4]: the output of automatic normalization is a (well constrained) set of legitimate transliterations for each Old Hungarian token, from which a human annotator can select the context fitting element. The paper will present how classic NLP paradigms can be utilized to aid the creation of such a tool. Section 2 will give a brief overview of related work, section 3 will describe

the noisy channel model as applied to the normalization task while section 4 will give details of the training process. In section 5 we present the application of the model with experiments and evaluation. Conclusions and suggestion for further work will end the paper in section 6.

2 RELATED WORK

Depending on the approach to be taken in searching for a solution for the normalization problem, several "metaphors" can be considered, which are succinctly discussed in [5], so we only give here a short mention of the most important ones. A natural analogy to apply is considering the normalization task as a translation from one representation language to another and therefore using machine translation models [6, 7, 8]. Most of the objections that can be raised against using this approach for our purposes are already mentioned in [5]: given the relative similarity of the "source" and "target" language in this case launching the complex machinery of MT models might be unnecessary; modelling word order mismatch and other complex relationships between source and target sentences seems almost an overkill.

Another related family of conversion tasks is grapheme-to-phoneme conversion where most of the solutions are based on the early probabilistic model of [9]. A detailed comparison of possible models can be found in [10], where it is shown that machine learning approaches in general perform better than models based upon handwritten rules. This problem has also been tackled in other popular learning paradigms including analogical and hidden Markov models [11, 12].

Most of the work on text normalization of historical documents is centered around a manually crafted or automatically induced set of correspondence rules [13], some form of approximate matching based on a distance metric (often the Levenshtein distance or some of its variant) [14], or some hybrid method combining the above two and sometimes assigning probabilities to the induced rules [15, 16]. However, the approach as taken here is most closely related to the spell-checking problem, which has been extensively researched in the past, originating from the method of [17]. A very successful class of current solutions are defined in the noisy channel paradigm [18, 19], which the next section will describe in detail, with respect to the normalization task. There is no denying that a number of further machine learning techniques could be applied to find an efficient solution to the problem, from decision tree to log-linear classifiers, however, given some practical constraints and requirements coming from the project (the model should be easy to formulate, simple and quick to implement and integrate into a chain of modules responsible for different steps of normalization; perform well enough to be a significant aid in (the inevitable) manual correction process), we opted for the approach described in the next Section.[2]

3 NOISY CHANNEL TEXT NORMALIZATION

Following standard practice in noisy channel modelling we consider an original Old Hungarian string as an observation of the modern normalized version that has been sent through a noisy communication channel (Figure 1). Let M be a character (sub)string from the normalized version of a sentence in modern spelling, and let O denote the original Old Hungarian transcription. The task of the decoder to find the character string $\hat{M}$ for which the $P(M|O)$ conditional probability maximal:

$$\hat{M} = \underset{M}{\operatorname{argmax}}\, P(M|O) \tag{1}$$

With the standard application of Bayes' rule (1) becomes:

$$\hat{M} = \underset{M}{\operatorname{argmax}}\, \frac{P(O|M)P(M)}{P(O)} = \underset{M}{\operatorname{argmax}}\, P(O|M)P(M) \tag{2}$$

The task is now formulated in terms of developing models for the two component distributions: the $P(O|M)$ transliteration model-distribution (channel model) and the $P(M)$ normalized textmodel-distribution (source model).

Utilizing the simplest of potential frameworks, the source model can be built as a character n-gram model from the normalized text in modern spelling. This normalized text being practically identical to contemporary Hungarian (with sporadic occurrences of some Old Hungarian morphological phenomena), a more or less unlimited amount of training data is available to build the source model, so relatively high order models ($n = 4, n = 5$) can also be considered. To estimate the parameters of the transliteration model several methods can be considered, with the prerequisite of a training set of $M_i^j \rightarrow O_k^l$ mappings between original and modern substrings.[3] To represent contextual information in the transliteration model mappings between strings longer than one character could also be defined. The training set can then be used to build the transliteration model while the set of possible modern variants for each Old Hungarian token could be generated from the mappings. This approach is essentially the same as the one in [18] so a similar formal description of the transliteration model can be given along their lines.

Let Part(M) be the set of all possible partitions from adjacent substrings of the modern normalized string; similarly Part(O) for the original form. For a given $R \in$ Part(M) partition, where R consists of $|R| = j$ segments let R_i denote the i^{th} segment. Now given $|T| = |R|$ where $T \in$ Part(O) the $P(O|M)$ conditional probability can be calculated:

$$P(O|M) = \sum_{R \in \text{Part}(M)} P(R|M) \sum_{T \in \text{Part}(O)} \prod_{i=1}^{j} P(T_i|R_i) \tag{3}$$

One particular alignment will correspond to a specific set of $M_i^j \rightarrow O_k^l$ mappings. If we consider only the best partitioning (3) could be simplified to:

$$P(O|M) = \max_{R \in \text{Part}(M), T \in \text{Part}(O)} P(R|M) \prod_{i=1}^{j} P(T_i|R_i) \tag{4}$$

Similarly to [18] in the current model we also drop the $P(R|M)$ component (i.e. in lack of any better way of determining $P(R|M)$ we assume a uniform distribution over the possible partitions).

4 TRAINING THE MODEL

4.1 Training data for the transliteration model

Source for the training data consists of 24 short Old Hungarian texts and a part of an Old Hungarian codex manually normalized by historical linguists. The earliest Hungarian sources usually are short parts of Latin codices. Most of them are pieces of religious literature, but secular texts were also created in the late Old Hungarian

[2] In a different vein there has already been some related work in Hungarian with the rule based model of [20].

[3] $i < j,\ k < l$ are indexes specifying positions in between characters, so if $j = i + 1,\ l = k + 1$ we get a one-to-one character→character mapping.

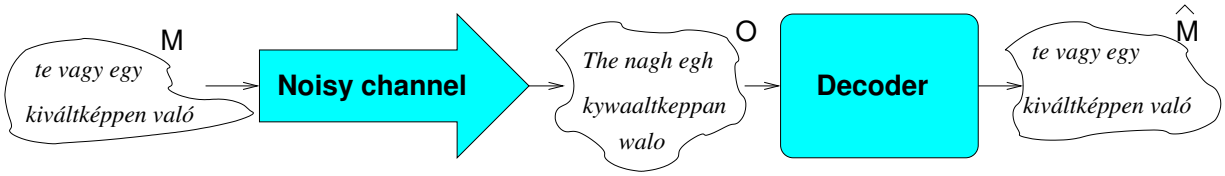


Figure 1. Text normalization in the noisy channel model

period. The oldest text in train data is the *Halotti beszéd és könyörgés* ('Funeral Oration and Prayer'), which is the first written document in the whole Uralic language family from about 1195. The youngest one in the training data is from the last year of Old Hungarian period, 1526 — it is a versed chronicle about the purchase of Pannonia. The codex part is from the *Székelyudvarhelyi kódex* ('Codex from Székelyudvarhely'), which is a Franciscan codex written for nuns in 1526–28. Considering all available output from manual normalization the number of word tokens that could be used for the training data set approaches 20.000.

The training set actually used for the model consists of about 200 default $M_i^j \rightarrow O_k^l, j = i+1,\ l = k+1, j = l$ one-to-one equivalent mappings and mappings for strings not equal in length where there is an empty symbol at either side. Default mappings are extended with contextual ones where up to N adjacent symbols are concatenated to each side. With $N = 2$, approximately 2700 extensions are added to the default mapping set. Mappings containing only an empty symbol on either side are not considered, this fact is only indirectly represented in the extension set. To give an example, with $N = 2, M = \text{te}, O = \text{the}$, the following default mappings are generated (of these, the second mapping will be disregarded):

$$\begin{array}{ccc} \text{t} & \rightarrow & \text{t} \\ \epsilon & \rightarrow & \text{h} \\ \text{e} & \rightarrow & \text{e} \end{array}$$

The extension set will contain the following new rules:

$$\begin{array}{ccc} \text{t} & \rightarrow & \text{th} \\ \text{e} & \rightarrow & \text{he} \\ \text{te} & \rightarrow & \text{the} \end{array}$$

The manual development of the training set is supported by a simple semi-automatic procedure with some basic heuristics to find the initial alignment. The alignment algorithm first produces mappings between similar letter symbols in similar token internal position. For symbols that could not be aligned in the first run a small window of adjacent symbols is examined to generate candidate mappings. This way, manual work is reduced to the correction of the initial alignment.[4]

From the training set the probability of mappings can be calculated straightforwardly:

$$P(\alpha \rightarrow \beta) = \frac{C(\alpha \rightarrow \beta)}{C(\alpha)} \tag{5}$$

where $C(\alpha \rightarrow \beta)$ is the number of $\alpha \rightarrow \beta$ replacements in the training set and $C(\alpha)$ is the number of occurrences of the α string.

[4] Clearly, a more principled model generating the initial alignments could be considered based on for example [21].

4.2 The source model

The source model is generated from an approximately 10 million word, 65 million character subcorpus of the Hungarian National Corpus. In this magnitude differences in genre/register of texts are negligible and have little effects on the parameters of the model. Likewise, the specific smoothing technique applied for the model has no real consequence on the performance of the application, so following standard practice we used the CMU toolkit [22] to develop the model with the default Good-Turing smoothing (altering this setting had no effect on the results so the default selection was retained all throughout testing).

5 APPLYING THE MODEL

For a given O original string the value of $\text{argmax}_M\ P(O|M)P(M)$ has to be calculated. In the current settings this is done in the following (not the most optimal[5]) way: according to the mappings in the transliteration model an n-best list from all possible forms from all possible partitions[6] are generated, together with probabilities assigned from the transliteration model. This candidate list is then reordered by using the probabilities assigned from the source model, producing the final output of the algorithm. In spelling correction the generation of candidates is normally dictionary/lexicon constrained, here, Hungarian being a highly inflectional language with the number of possible wordforms out of reach for any lexical list, the natural choice for representing the vocabulary can be a morphological analyzer (MA). Still, the highest scoring OOV items should also be considered as possible outputs since the MA lexicon does not have full coverage of the Old Hungarian vocabulary (testing the OOV ratio on about 20.000 manually normalized tokens and 4500 types yielded 14.5% OOV tokens and 19.8% OOV types, which have to be taken care of by manual effort).

The requirement of returning only the highest scoring normalized candidate is not practical in the context of this task. Very often there are several equally legitimate modern equivalents for a particular original form (see Figure 2 for an example) which are impossible to disambiguate at this level of processing; of these only a human annotator equipped with her native speaker competence will be able to select the right solution.

Nonetheless, we built a reference/supplementary decision tree learner which inherently only provides one output. The character-based learner used the five characters in a two characters wide window around the current character as context features. Thus, for every Old Hungarian character, the decision tree determines the most likely modern character based on the context. C4.5 decision trees were built using the WEKA machine learning package [23]. The (MA filtered)

[5] Standard optimization techniques such as beam search pruning can naturally be applied but have not yet been implemented in this testing phase.

[6] The maximum length of a string in a partition is $N+1$, i.e. equals the length of the longest string in the extended mappings.

$$toroknek \Rightarrow \begin{cases} \textit{töröknek} \text{ 'Turk (dative)'} \\ \textit{tőröknek} \text{ 'daggers (dative)'} \\ \textit{tőrüknek} \text{ 'their dagger (dative)'} \end{cases}$$

Figure 2. Multiple normalizations for one input

output of the decision tree module is used to reorder the selection list from the channel module: if this output is found among the first ten recommendations of the MA filtered channel module it is promoted as the highest candidate offered for normalization. This gives us the schematic architecture of the present system in Figure 3, where neither the MA nor the decision tree module prunes the selection list only reorders it.

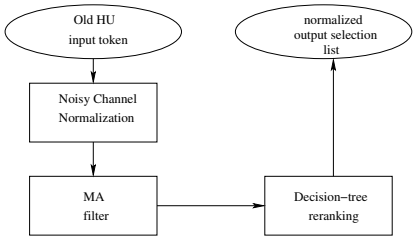


Figure 3. The normalization architecture

5.1 Evaluation

For the decision-tree learner a train/test split of about 45/55 percent was applied to a 7500 word data set. Two different settings of the confidence factor (CF), which is used by the decision tree learning algorithm to shape the degree of pruning, yielded the following results:

model	accuracy
CF = 0.25 (more pruning)	64.38%
CF = 0.5 (less pruning)	69.29%

Table 1 shows some illustrative examples where the method worked flawlessly:

For the noisy channel model, in accordance with the requirements of the context of the actual task, the most appropriate evaluation framework is the n-best list evaluation, which is presented here for values of $n = 1, 3, 5, 10, 15, 20$. The n-best accuracy reported here is the percentage of time the correct form is present in the top n candidates proposed by the system. The training set here was generated from 1100 Old HU tokens (≈6400 characters) and their normalized versions while the test set contained 200 randomly selected Old HU word types (≈1100 characters)[7]. Figure 4 illustrates the output of the method without MA filtering while Table 5.1 presents the precision values for the different n settings.

[7] This is admittedly a low number due to the current inefficient implementation. Still it can give a good indication about the overall performance of the method.

Table 1. Correct suggestions of the decision-tree learner

Task	Old HU	modern HU	gloss
handling *ű*	*alaűa* *űrasnak* *űl'*	*alája* *írásnak* *ily*	below sg writing (dative) such
handling *w*	*hwllatwan*	*hullatván*	sheding
where to put accents	*varosaban* *űrgalmassagot*	*városában* *irgalmasságot*	his city (iness.) mercy (acc.)
variations of a word	*vereé* *veréé*	*veré* *veré*	beat (3sg, past, def.) beat (3sg, past, def.)
complex example	*bwzzwth*	*bosszút*	revenge (acc.)

```
fwl (fül (ear))=>
-8,80780895229285       föl
-10,7227286786192       fel
-11,0558158154337       fül
-11,2756412387919       fől
-12,4574295350367       fol*

honneg (honnét (from where))=>
-19,1117218113907       honneg*
-19,5230300429664       honnég*
-20,8376176340216       honnét
-21,8538140705439       honyneg*
-22,5639991398073       hónneg*

ygen (igen (yes))=>
-10,8729908279143       igén
-11,3178857141749       igen
-11,5989613202567       igény
-13,4229320257043       igyen*
```

Figure 4. n-best list outputs for various inputs ('*' marks ill-formed tokens)

Table 2. Precision results for the three (raw noisy channel, MA filtered and DT reranked) architectures

	Precision					
	1-best	3-best	5-best	10-best	15-best	20-best
Raw NC	37.7	64.4	74.3	84.3	86.9	87.9
MAF	62.3	80.1	83.8	87.4	87.4	87.9
DTR	73.3	82.2	84.3	87.4	87.4	87.9

Compared to results from the spelling correction literature (eg. [19]) these values are significantly lower, however, given the difficulty of the task (much higher variability in the input and the conversion) this simple method is already offering promising perspectives for practical use in the project. The results nicely reflect the beneficial effect of the gradual extension of the information sources through combining the outputs from the different classifiers and prove that a workable solution can be found along the presented lines.

6 CONCLUSION

The paper discussed how a simple stochastic model with some straightforward extensions can be applied to the normalization of historical texts as a potential framework from many to translate between two types of representations. It is undeniable that for an exhaustive investigation of the normalization problem several further machine learning approaches mentioned in section 2 could be taken into account [24, 25, 12], however, the approach presented here can already give a usable solution for the purposes of the project and significantly cut back the manual effort necessary to create high quality resources.

One interesting line of future research for still improving performance is the combination of additional types of classifiers to add even more support to the human decision process over the set of potential candidates. One promising possibility could be to use a log-linear classifier such as a maximum entropy model, where a wide scale and type of features can easily be taken into account (eg. capitalization, position of a particular character within the word etc.). A lot more extensive and detailed evaluation is also needed to give more insights on the usability and behavior of the method, in particular its scalability is a crucial factor: how much of a codex should be manually tagged to train a system with acceptable accuracy on the output. On the practical side, some further programming work is required until an efficient and user friendly implementation of the whole toolset is developed and becomes available for the project to start with the large scale development of historical corpora.

ACKNOWLEDGEMENTS

This research is supported by the Hungarian Scientific Research Fund (OTKA grant NK-78074). The authors are grateful to the anonymous reviewers for their valuable comments, which were very helpful in preparing the final version of this paper.

REFERENCES

[1] A. Kroch, B. Santorini, and L. Delfs. Penn-Helsinki parsed corpus of Early Modern English, 2004. http://www.ling.upenn.edu/hist-corpora/PPCEME-RELEASE-1/.

[2] P. Rayson, D. Archer, A. Baron, J. Culpeper, and N. Smith, 'Tagging the Bard: Evaluating the accuracy of a modern POS tagger on Early Modern English corpora', in *Proceedings of Corpus Linguistics*, University of Birmingham, (2007).

[3] C. E. Shannon, 'A mathematical theory of communication', *Bell System Technical Journal*, **27**(3), 379–423, (1948).

[4] A. Baron and P. Rayson, 'VARD 2: A tool for dealing with spelling variation in historical corpora', in *Proceedings of the Postgraduate Conference in Corpus Linguistics*, Aston University, Birmingham, (2008).

[5] F. Yvon, 'Rewriting the orthography of SMS messages', *Natural Language Engineering*, **16**(2), 133–159, (2010).

[6] K. Raghunathan and S. Krawczyk, 'Investigating SMS text normalization using statistical machine translation'. Stanford University, 2009.

[7] C. Kobus, F. Yvon, and G. Damnati, 'Normalizing SMS: are two metaphors better than one?', in *Proceedings of the 22nd International Conference on Computational Linguistics*, volume 1, pp. 441–448, Manchester, United Kingdom, (2008). Association for Computational Linguistics.

[8] A. Aw, M. Zhang, J. Xiao, and J. Su, 'A phrase-based statistical model for SMS text normalization', in *Proceedings of the COLING/ACL*, pp. 33–40, Sydney, Australia, (2006). Association for Computational Linguistics.

[9] J. Lucassen and R. L. Mercer, 'An information theoretic approach to the automatic determination of phonemic baseforms', in *ICASSP-84*, volume 9, pp. 304–307, (1984).

[10] R. I. Damper, Y. Marchand, M. J. Adamson, and K. Gustafson, 'Evaluating the pronunciation component of text-to-speech systems for English: A performance comparison of different approaches', *Computer Speech and Language*, **13**(2), 155–176, (1999).

[11] J. R. Bellegarda, 'Unsupervised, language-independent grapheme-to-phoneme conversion by latent analogy', *Speech Communication*, **46**(2), 140–152, (2005).

[12] P. Taylor, 'Hidden Markov models for grapheme to phoneme conversion', in *INTERSPEECH-05*, pp. 1973–1976, Lisbon, Portugal, (2005).

[13] T. Pilz, A. Ernst-Gerlach, S. Kempken, P. Rayson, and D. Archer, 'The identification of spelling variants in english and german historical texts: manual or automatic?', *Literary and Linguistic Computing*, **23**(1), 65–72, (2008).

[14] J. Strunk, 'Information retrieval for languages that lack a fixed orthography', Technical report, Linguistics Department, Stanford University, (2003).

[15] T. Pilz, W. Luther, and N. Fuhr, 'Rule-based search in text databases with nonstandard orthography', *Literary and Linguistic Computing*, **21**(2), 179–186, (2006).

[16] A. Ernst-Gerlach and N. Fuhr, 'Generating search term variants for text collections with historic spellings', in *Advances in Information Retrieval*, eds., M. Lalmas, A. MacFarlane, S. Rüger, A. Tombros, T. Tsikrika, and A. Yavlinsky, 49–60, Springer, (2006).

[17] M. D. Kernighan, K. W. Church, and W. A. Gale, 'A spelling correction program based on a noisy channel model', in *COLING-90*, volume II, pp. 205–211, Helsinki, (1990).

[18] E. Brill and R. C. Moore, 'An improved error model for noisy channel spelling correction', in *ACL-00*, pp. 286–293, Hong Kong, (2000).

[19] K. Toutanova and R. C. Moore, 'Pronunciation modeling for improved spelling correction', in *ACL-02*, pp. 144–151, Philadelphia, PA, (2002).

[20] G. Kiss, M. Kiss, and J. Pajzs, 'Normalisation of hungarian archaic texts', in *Proceedings of COMPLEX 2001*, pp. 83–94. University of Birmingham, (2001).

[21] A. W. Black, K. Lenzo, and V. Pagel, 'Issues in building general letter to sound rules', in *3rd ESCA Workshop on Speech Synthesis, Jenolan Caves, Australia*, (1998).

[22] P. R. Clarkson and R. Rosenfeld, 'Statistical language modeling using the CMU-Cambridge toolkit', in *EUROSPEECH-97*, volume 1, pp. 2707–2710, (1997).

[23] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, 'The WEKA data mining software: An update', *SIGKDD Explorations*, **11**(1), (2009).

[24] S. F. Chen, 'Conditional and joint models for grapheme-to-phoneme conversion', in *EUROSPEECH-03*, (2003).

[25] Y. Marchand and R. I. Damper, 'A multi-strategy approach to improving pronunciation by analogy', *Computational Linguistics*, **26**(2), 195–219, (2000).

Reducing OCR Errors by Combining Two OCR Systems

Martin Volk and **Torsten Marek** and **Rico Sennrich** [1]

Abstract. This paper describes our efforts in building a heritage corpus of Alpine texts. We have already digitized the yearbooks of the Swiss Alpine Club from 1864 until 1982. This corpus poses special challenges since the yearbooks are multilingual and vary in orthography and layout. We discuss methods to improve OCR performance and experiment with combining two different OCR programs with the goal to reduce the number of OCR errors. We describe a merging procedure that uses a unigram language model trained on the uncorrected corpus itself to select the best alternative, and report on evaluation results which show that the merging procedure helps to improve OCR quality.

1 INTRODUCTION

In the project Text+Berg[2] we digitize the heritage of alpine literature from various European countries. Currently our group digitizes all yearbooks of the Swiss Alpine Club (SAC) from 1864 until today. Each yearbook consists of 300 to 600 pages and contains reports on mountain expeditions, culture of mountain peoples, as well as the flora, fauna and geology of the mountains.

Some examples from the 1911 yearbook may illustrate the diversity. There are the typical reports on mountain expeditions: "*Klettereien in der Gruppe der Engelhörner*" (English: *Climbing in the Engelhörner group*) or "*Aus den Hochregionen des Kaukasus*" (English: *From the high regions of the Caucasus*). But the 1911 book also contains scientific articles on the development of caves ("*Über die Entstehung der Beaten- und Balmfluhhöhlen*") and on the periodic variations of the Swiss glaciers ("*Les variations périodiques des glaciers des Alpes suisses*").

The corpus is thus a valuable knowledge base to study the changes in all these areas. But the corpus is also a resource to catch the spirit of Switzerland in cultural terms: What does language use in alpine texts show about the cultural identity of the country and its change over time?[3]

This paper describes the corpus and the project phases from digitalization through annotation to publication. We focus on the language technology challenges in improving optical character recognition (OCR). More details on the other project phases can be found in [10].

2 THE TEXT+BERG CORPUS

The Swiss Alpine Club was founded in 1863 as a reaction to the foundation of the British Alpine Club a year before. Thus our corpus has a clear topical focus: conquering and understanding the mountains. The articles focus mostly on the Alps, but over the 145 years the books have probably covered every mountain region on the globe.

The corpus is multilingual. Initially the yearbooks contained mostly German articles and few in French. Since 1957 the books appeared in parallel German and French versions (with some Italian articles) which allows for interesting cross-language comparisons and may serve as training material for Statistical Machine Translation systems.

3 PROJECT PHASES

We have collected all books in two copies (as a result of a call for book donations by the Swiss Alpine Club). One copy was cut open so that the book can be scanned with automatic paper feed. The other copy remains as reference book.

In a first step, all books are scanned and processed by OCR systems. The main challenges for OCR we encountered are the multilingual nature of the text, diachronic changes in spelling and typesetting, and the wide range of proper nouns. In section 4, we will focus on our ongoing efforts to improve OCR quality.

After text recognition we add a mark-up of the text structure. Specially developed programs annotate the text with TEI-conformant XML tags for the beginning and end of each article, its title and author, subheaders and paragraphs, page breaks, footnotes and caption texts. For example, footnotes are recognized by their bottom position on the page, their smaller font size and their starting with any character followed by a closing parenthesis.

Some of the text structure information can be checked against the table of contents and table of figures in the front matter of the yearbooks. We manually correct these tables as the basis for a clean database of all articles in the corpus. Matching entries from the table of contents to the articles in the books is still not trivial. It requires that the article title, the author name(s) and the page number in the book are correctly recognized. Therefore, we use fuzzy matching to allow for OCR errors and small variations between table of content entries and the actual article header in the book.

3.1 Language identification

Proper language identification is important for most steps of automatic text analysis, e.g. part-of-speech tagging, lemmatization and named entity classification. The SAC yearbooks are multilingual, with most articles written in German and French, but also some in Italian, Romansch and Swiss German[4]. We use a character-n-gram-based language identification program[5] to determine the language for each sentence.

[1] University of Zurich, Switzerland, email: [lastname]@cl.uzh.ch
[2] See www.textberg.ch.
[3] See [3] for our research in this area.
[4] See section 3.3 for information on the amount of text in each language.
[5] We use Michael Piotrowski's language identifier Lingua-Ident from search.cpan.org/dist/Lingua-Ident/ .

Table 1. Token counts (rounded) in the Text+Berg corpus

	German	French	Italian	English	**Total**
tokens in entire corpus	17,253,000	8,126,000	329,000	44,000	25,753,700
tokens in parallel subcorpus	2,295,000	2,604,000			

While language identification may help improve automatic text analysis, the dependency is circular. OCR, tokenization and sentence boundary recognition need to precede language identification so that we are able to feed individual sentences to the language identifier. But high quality tokenization relies heavily on language-specific abbreviation lists and conventions. We therefore perform an initial tokenization and sentence boundary recognition before language identification. Afterwards, we retokenize the text in order to correct possible tokenization errors.

OCR is performed without prior language identification. We configured the OCR systems to use the dictionaries for the following four languages: German, French, Italian and English.

3.2 Further annotation

Apart from structural mark-up and language identification, the corpus is automatically tagged with POS information. We also aim to provide a fine-grained annotation of named entities.

Named entity recognition is an important aspect of information extraction. But it has also been recognized as important for the access to heritage data. [2] argue for named entity recognition in 19th century Swedish literature, distinguishing between 8 name types and 57 subtypes.

In the latest release of our corpus, we have annotated all mountain names that we could identify unambiguously through exact matching. We have obtained a large gazetteer with 156,000 toponyms from the Swiss Federal Office of Topography. It contains geographical names in 61 categories. We have extracted the SwissTopo mountain names from the 4 highest mountain classes plus the names classified as ridges (*Grat*). This resulted in 6227 names from which we have manually excluded 50 noun homographs. For example *Ofen* (English: *oven*) is a Swiss mountain name, but in order to avoid false hits we eliminated it from the list. The resulting gazetteer triggered the identification of 66,500 mountain names in our corpus.

3.3 Aims and current status

In the final phase, the corpus will be stored in a database which can be searched via the internet. Because of our detailed annotations, the search options will be more powerful and lead to more precise search results than usual search engines. For example, it will be possible to find the answer to the query "List the names of all glaciers in Austria that were mentioned before 1900." We also annotate the captions of all photos and images so that they can be included in the search indexes.

[11] emphasize that advanced access methods are crucial for Cultural Heritage Data. They distinguish different user groups having different requirements (Historians, Practitioners, Laypersons, Computational Linguists). We will provide easy access to the texts and images through a variety of intuitive and appealing graphical user interfaces. We plan to have clickable geographic maps that lead to articles dealing with certain regions or places.

As of June 2010, we have scanned and OCR-converted 142 books from 1864 to 1982. We have 90 books from 1864 to 1956. In 1870, 1914 and 1924 no yearbooks were published. From 1957 to 1982 we have parallel French and German versions of the yearbooks. Overall we have scanned nearly 70,000 pages. The corpus is made up of 6101 articles in German, 2659 in French, 155 in Italian, 12 in Romansch, and 4 in Swiss-German. This count includes duplicates from the parallel French and German yearbooks. 458 of the articles in these parallel books are not translated but reprinted in the same language. This means we currently have a corpus of 8931 articles in one of the languages. Our parallel corpus currently contains 701 articles amounting to 2.6 million tokens in French and 2.3 million tokens in German. Table 1 gives an overview of the token frequencies per language. Work on scanning and converting the yearbooks from 1983 is ongoing and will be finished later this year.

4 SCANNING AND OCR

We started by using Abbyy-FineReader 7[6], a state-of-the-art OCR software to convert the images to text. This software comes with two lexicons for German, one for the spelling after 1901 and one for the new orthography following the spelling reform of the late 1990s. The system does not have a lexicon for the German spelling of the 19th century (e.g. old *Nachtheil*, *passiren* and *successive* instead of modern *Nachteil*, *passieren* and *sukzessive*). We have therefore added 19th century word lists to the system. We have manually corrected one book from 1890, and subsequently extracted all words from that book that displayed old-style character sequences (such as 'th', 'iren', and 'cc').

The 20th century books follow the Swiss variant of German spelling. In particular, the Swiss spelling has abandoned the special character 'ß' in favor of 'ss'. For example, the word *ließ* (English: *let*) is spelled *liess* in Switzerland. The OCR lexicons list only the spelling from Germany. We have therefore compiled special word lists with Swiss spelling variants taken from the GNU Aspell program.

Names that are not in the system's lexicon pose another problem to character recognition. Our books contain a multitude of geographical names many of which are unknown to the OCR system. We have therefore purchased a large list of geographical names from the Swiss Federal Office of Topography (www.swisstopo.ch) and extracted the names of the major Swiss cities, mountains, valleys, rivers, lakes, hiking passes and mountain cabins. In total we added 14,800 toponyms to the OCR system.

These steps have helped us to further improve the OCR quality, which was good from the very start of the project, thanks to the fact that the yearbooks were set in Antiqua font from the first publishing year 1864. So we do not have to deal with old German Gothic font (Fraktur).

A group of student volunteers helps in the correction of the automatically recognized text. The idea is to get the text structure right and to eliminate the most obvious OCR errors. Additionally, we post-correct errors caused by graphemic similarities which have been

[6] We have initially evaluated Abbyy-FineReader version 7 to version 9, but found the older version more stable and of equal OCR quality.

missed by the OCR engine. This automatic correction happens after tokenization on a word by word level, using a rule-based approach. For example, a word-initial 'R' is often misinterpreted as 'K', resulting in e.g. *Kedaktion* instead of *Redaktion* (English: *editorial office*). To minimize false positives, our rules fall in one of three categories: First, strict rule application: The tentative substitute must occur in the corpus and its frequency must be at least 2.5 times as large as the frequency of the presumably mistyped word, and the suspect must not occur in the German newspaper corpus TIGER. Second, normal rule application: The tentative substitute must occur in the corpus. Third, unconditional substitution. The above *K→R* example falls in the strict category; substituting 'ii' by either 'n', 'u', 'ü', 'li' or 'il' (of the five tentative substitutes the word with the highest frequency is selected; e.g. *iiberein* → *überein*, English: in agreement) falls in the normal category; and substituting *Thai* with *Thal* (the 19th century spelling of *Tal*, English: *valley*) is an example of the unconditional rule category.

4.1 OCR merging

Even though the performance of OCR applications is satisfactory for most purposes, we are faced with thousands of OCR errors in large text collections. Since we aim to digitize the data as cleanly as possible, we naturally wish to minimize the number of errors. Additionally, OCR errors can be especially damaging for some applications. The numerous named entities, i.e. names of mountains, streams and Alpine cabins are especially prone to OCR errors, especially because many of them do not occur in the dictionaries used by OCR tools. At the same time, these named entities are highly relevant for our goal of building a searchable database.

In our project, OCR is complicated by the fact that we are digitizing a multilingual and diachronic corpus, the texts spanning from 1864–1982. We have evaluated different OCR systems, and we are continuing our efforts to better adapt them to our collection of Alpine texts, for instance by adding lists of mountain names and orthographic variants to the OCR dictionaries.

In an attempt to automatically detect and correct the remaining OCR errors, we exploit the fact that different OCR systems make different errors. Ideally, we can eliminate all OCR errors that are only made by one of two systems. We have created an algorithm that compares the output of two OCR systems (Abbyy FineReader 7 and OmniPage 17) and performs a disambiguation, returning the top-ranking alternative wherever the systems produce different results.

4.2 Related work

Other methods for automatic OCR-error correction include e.g. statistical approaches as described in [6] and [4] as well as lexical approaches as in [9]. As for the combination of multiple OCR systems, research has identified two main questions: how to efficiently align the output of multiple OCR systems (e.g. [5]), and how to select the optimal word among different candidates. The latter step is performed using voting algorithms [8], dictionaries [5], or human post-editing [1].

4.3 Algorithm

For our task, we can avoid potential complexity problems since we do not have to compute a global alignment between the two OCR systems. Three factors help us keep the search space small: Firstly, we can extract differences page-by-page. Secondly, we ignore any differences that cross paragraph boundaries, defaulting to our primary system FineReader if such a large discrepancy should occur. Thirdly, the output of the two systems is similar enough that differences typically only span one or two words.

For each page, the algorithm traverses the two OCR-generated texts linearly until a difference is encountered. This point is then used as starting point for a longest common subsequence search in a 40-character-window. We extract as difference everything up to the start of the longest subsequence, and continue the algorithm from its end.

For selecting the best alternative, we consider the differences on a word level. If there are several differences within a short distance, all combinations of them are considered possible alternatives. As a consequence, we not only consider the output of FineReader (*Recensione-»,*) and OmniPage (*Rccensionen*), but also the combinations *Rccensione-»,* and *Recensionen*. In this way, the correct word form *Recensionen* can be constructed from two wrong alternatives.

Our decision procedure is based on a unigram language model trained on the latest release of the Text+Berg corpus. The choice to bootstrap the decision procedure with noisy data generated by Abbyy FineReader bears the potential risk of skewing the selection in Abbyy FineReader's favor. However, the language model is large (25.7 mio words), which means that possible misreadings of a word are far outnumbered by the correct reading. For instance, *Bergbauer* (English: *mountain farmer*) is twice misrecognized as *ßergbauer* by Abbyy FineReader. Still, *Bergbauer* is more than 20 times as frequent as *ßergbauer* in the corpus (47 vs. 2 occurrences), which lets the language model make a felicitous judgment.

It is worth noting that OCR merging is performed before language identification, and that we do not use one model per language, but a language model trained on the whole corpus, irrespective of language.

Words containing non-alphabetical characters have been removed from the language model, with the exception of hyphenated words. Punctuation marks and other special characters are thus penalized in our decision module, which we found to be an improvement.

A language model approach is problematic for cases in which the alternatives are tokenized differently. Generally, alternatives with fewer tokens obtain a higher probability. We try to counter this bias with a second score that prefers alternatives with a high ratio of known words. This means that *in Göschenen* is preferred over *in-Göschenen*, even if we assume that both *Göschenen* (the name of a village) and *inGöschenen* are unknown words in our language model[7].

The alternatives are ranked first by the ratio of known words, second by their language model probability. If there are several candidates with identical scores, the alternative produced by Abbyy FineReader is selected.

4.4 Results

We have manually corrected OCR errors in the Swiss Alpine Club yearbook 1899, starting from the Abbyy FineReader output, which we consider our primary system because of its better performance. This book, spanning 488 pages and containing approximately 220,000 tokens, serves as a gold standard for our OCR evaluation. Using The ISRI OCR Performance Toolkit [7], we measure 1260 word errors by Abbyy FineReader, and 6466 by OmniPage, yielding word accuracies of 99.26% and 96.21%, respectively, as table

[7] Unknown words are assigned a constant probability > 0.

Table 3. Examples where OmniPage is preferred over FineReader by our merging procedure.

Abbyy FineReader	OmniPage	correct alternative in context	jugdment
Wunseh,	Wunsch,	entstand in unserem Herzen der **Wunsch**,	better
East	Rast	durch die **Rast** neu gestärkt	better
Übergangspunkt,. das	Übergangspunktr das	ist Hochkrumbach ein äußerst lohnender Übergangspunkt, das	equal
großen. Freude	großen, Freude	zu meiner **großen Freude**	equal
halten	hatten	Wir **halten** es nicht mehr aus	worse
là	la	c'est **là** le rôle principal qu'elle joue	worse

Table 2. Word accuracy of different OCR systems.

system	word errors	word accuracy
Abbyy FineReader 7	1260	99.26%
OmniPage 17	6466	96.21%
merged system	1305	99.24%

2 shows. The merged system performs slightly worse, with 1305 misrecognized words and an accuracy of 99.24%.[8] These negative results led us to manually investigate an 8-page article with 9 differences between Abbyy FineReader and the merged system. According to the gold standard, the merge increases the number of word errors from 8 to 15, which means that 8 out of 9 modifications are errors. However, a manual analysis of these 8 reported errors shows that half of them are actually improvements, with the error being in the gold standard.

The task of creating a gold standard for this 488-page book was very time-consuming, and we deem its quality sufficiently high to identify large quality differences between systems. However, FineReader is slightly advantaged by the fact that the gold standard is based on its output; when evaluating FineReader, every uncorrected error in the gold standard will count as correct. Conversely, correcting an error that is not corrected in the gold standard is penalized in the automatic evaluation. We have thus performed an alternative, manual evaluation of the merged algorithm based on all instances where the merged system produces a different output than Abbyy FineReader. The cases where Abbyy's system wins are not as interesting since we regard them as the baseline result. Out of the 1800 differences identified between the two systems[9] in the 1899 yearbook, the FineReader output is selected in 1350 cases (75%); in 410 (23%), the OmniPage reading is preferred; in 40 (2%), the final output is a combination of both systems. We manually evaluated all instances where the final selection differs from the output of Abbyy FineReader, which is our baseline and the default choice in the merging procedure.

Table 3 shows some examples and our judgment. We see clear improvements where non-words produced by Abbyy FineReader (e.g. *Wunseh*) are replaced with a known word produced by OmniPage (*Wunsch*, English *wish*). On the other hand, there are cases where a correctly recognized Abbyy word (e.g. *halten*, English: *hold*) is overwritten by the OmniPage candidate (*hatten*, English: *had*) because the latter is more frequent in our corpus. As a third possibility, there are neutral changes where the Abbyy output is as wrong as the OmniPage output, as in the two examples judged as "equal", where the systems suggest different punctuation symbols where none is intended in the text.

The central question is whether the manual evaluation confirms the results of the automatic one, namely that our merging procedure does more harm than good, or whether there is actually an improvement. In our manual evaluation, we found 277 cases where OCR quality was improved, 82 cases where OCR quality was decreased, and 89 cases where combining two systems neither improved nor hurt OCR quality. This is in strong contrast to the automatic evaluation. While the automatic evaluation reports an increase in the number of errors by 45, the manual evaluation shows a net reduction of 195 errors.

We noticed that performance is worse for non-German text. Most notably, OmniPage tends to misrecognize the accented character *à*, which is common in French, as *A* or *a*, or to delete it. The misrecognition is a problem for words which exist in both variants, especially if the variant without accent is more common. This is the case for *la* (English: *the*) and *là* (English: *there*), and leads to a miscorrection in the example shown in table 3. We are lucky that in our language model, the French preposition *à* (English: *to*) is slightly more probable than the French verb *a* (English: *has*); otherwise, we would encounter dozens of additional miscorrections.[10] Word deletions are relatively rare in the evaluation set, but pose a yet unsolved problem to our merging algorithm. In 8 cases, *à* is simply deleted by OmniPage. These alternatives always obtain a higher probability than the sequences with *à*[11], and are thus selected by our merging procedure, even though the deletion is incorrect in all 8 instances.

Looking back at the automatic results, we estimate the number of errors by Abbyy FineReader to be between 1260 and 1800, allowing for up to one third of OCR errors being uncorrected in our gold standard. With a net gain of 200 corrections, we could correct about 10-15% of all errors. Considering that we are working with a strong baseline, we find it encouraging that using the output of OmniPage, which is considerably worse than that of Abbyy FineReader, allows us to further improve OCR performance.

5 CONCLUSION

We are working on the digitization and annotation of Alpine texts. Currently we compile a corpus of 145 German yearbooks and 52 French yearbooks from the Swiss Alpine Club. In the next step we will digitize the French yearbooks *L'Echo des Alpes* that were published in Switzerland from 1871 until 1924 to counterbalance the German language dominance in the yearbooks of the Swiss Alpine Club. We also have an agreement with the British Alpine Club to include their texts in our corpus.

[8] For the evaluation, the 1899 yearbook has been excluded from the language model used in our system combination procedure.

[9] Note that one difference, as defined by our merging algorithm, may span several words. Also, frequent differences that would be resolved in later processing steps (i.e. differences in tokenization or hyphenation) are ignored by the merging algorithm.

[10] Of course, one could devise rules to disallow particular corrections.

[11] Since every word has a probability < 1, each additional token decreases the total probability of an alternative.

We have shown that combining the output of two OCR systems leads to improved recognition accuracy at the current state of the art of the OCR systems. Surprisingly, these results were not visible in an automatic evaluation because of noise in the gold standard; only a manual investigation confirmed the improvement.

The fact that we now have the merging algorithm in place allows us to investigate the inclusion of further OCR systems. For this, Tesseract is an attractive candidate since it is open source and can thus be tuned to handle those characters well where we observe special weaknesses in the commercial OCR systems.

Our merging procedure also triggered further ideas for combining other textual sources. Our parallel French and German books since the 1950s contain many identical texts. These books are only partially translated, and they partially contain the same article in both books. We have already found out that even the same OCR system (Abbyy FineReader) makes different errors in the recognition of the two versions of the (same) text (e.g. *in der Gipfelfaüinie* vs. *inj der Gipfelfallinie*). This gives us more variants of the same text which we can merge.

We are also wondering whether the same text scanned under different scanner settings, e.g. different contrasts or different resolution, will lead to different OCR results which could be merged towards improved results. For instance, a certain scanner setting (or a certain image post-correction) might suppress dirt spots on the page which may lead to improved OCR quality.

Finally we would also like to explore whether translated texts can help in OCR error correction. Automatic word alignment might indicate implausible translation correspondences which could be corrected via orthographically similar, but more frequent aligned words.

ACKNOWLEDGEMENTS

We would like to thank the many student helpers who have contributed their time to this project. We are also grateful for the support by the Swiss Alpine Club and by Hanno Biber and his team from the Austrian Academy Corpus. Furthermore, we would like to thank the anonymous reviewers for their valuable comments. Part of this research has been funded by the Swiss National Science Foundation.

REFERENCES

[1] A. Abdulkader and M. R. Casey, 'Low cost correction of OCR errors using learning in a multi-engine environment', in *ICDAR '09: Proceedings of the 2009 10th International Conference on Document Analysis and Recognition*, pp. 576–580, Washington, DC, USA, (2009). IEEE Computer Society.

[2] L. Borin, D. Kokkinakis, and L.-J. Olsson, 'Naming the past: Named entity and animacy recognition in 19th century Swedish literature', in *Proceedings of The ACL Workshop on Language Technology for Cultural Heritage Data (LaTeCH 2007)*, Prague, (2007).

[3] N. Bubenhofer, *Sprachgebrauchsmuster. Korpuslinguistik als Methode der Diskurs- und Kulturanalyse*, de Gruyter, Berlin, New York, 2009.

[4] O. Kolak, W. Byrne, and P. Resnik, 'A generative probabilistic OCR model for NLP applications', in *NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pp. 55–62, Morristown, NJ, USA, (2003). Association for Computational Linguistics.

[5] W. B. Lund and E. K. Ringger, 'Improving optical character recognition through efficient multiple system alignment', in *JCDL '09: Proceedings of the 9th ACM/IEEE-CS joint conference on Digital libraries*, pp. 231–240, New York, NY, USA, (2009). ACM.

[6] M. Reynaert, 'Non-interactive OCR post-correction for giga-scale digitization projects', in *Proceedings of the Computational Linguistics and Intelligent Text Processing 9th International Conference, CICLing 2008*, ed., A. Gelbukh, Lecture Notes in Computer Science, pp. 617–630, Berlin, (2008). Springer.

[7] S. V. Rice, *Measuring the Accuracy of Page-Reading Systems*, Ph.D. dissertation, University of Nevada, 1996.

[8] S. V. Rice, J. Kanai, and T. A. Nartker, 'A report on the accuracy of OCR devices', Technical report, University of Nevada, (1992).

[9] C. M. Strohmaier, *Methoden der Lexikalischen Nachkorrektur OCR-Erfasster Dokumente*, Ph.D. dissertation, Ludwig-Maximilians-Universität, München, 2004.

[10] M. Volk, N. Bubenhofer, A. Althaus, M. Bangerter, L. Furrer, and B. Ruef, 'Challenges in building a multilingual alpine heritage corpus', in *Proceedings of LREC*, Malta, (2010).

[11] R. Witte, T. Gitzinger, T. Kappler, and R. Krestel, 'A semantic wiki approach to cultural heritage data management', in *Proceedings of LREC Workshop on Language Technology for Cultural Heritage Data (LaTeCH 2008)*, Marrakech, Morocco, (2008).

From Law Sources to Language Resources

Michael Piotrowski[1]

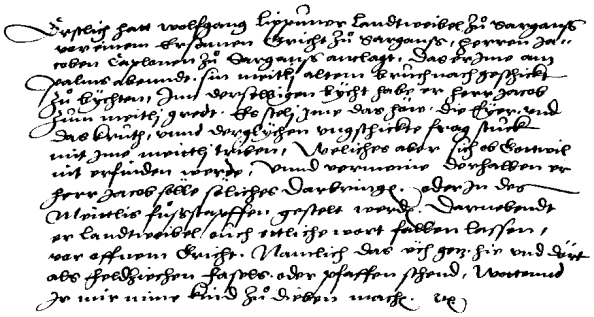
Abstract. The Collection of Swiss Law Sources is an edition of historical Swiss texts with legal relevance from the early Middle Ages up to 1798. The sources are manuscripts in historical variants of German, French, Italian, Rhaeto-Romanic, and Latin, which are transcribed, annotated, and published as editions of historical sources. The Collection is currently being digitized and will be made available on the Web as facsimiles. However, for a subset of the collection digital printing data in the form of FrameMaker documents is available. As this represents a sizable body of medieval and early modern text in various languages without OCR errors, it could serve as a valuable language resource for the processing of cultural heritage texts and for the development and evaluation of specialized NLP methods and techniques. This paper briefly describes the retrodigitization of the Collection of Swiss Law Sources and then discusses the conversion of the FrameMaker files in order to make the texts suitable for automatic processing and for the extraction and derivation of language resources.

1 INTRODUCTION

The Law Sources Foundation was founded in 1894 by the Swiss Lawyers Society in order to edit a collection of historical sources of law created on Swiss territory from the early Middle Ages up to 1798, the *Collection of Swiss Law Sources*. Since then, over 100 volumes, or more than 60,000 pages of source material (e.g., statutes, decrees, or regulations, but also administrative documents and court transcripts) from the early Middle Ages until early modern times have been published in the form of *source editions*. The primary sources are hand-written documents, which are transcribed, annotated, and commented by the editors; see figure 1 for an example of a source and its corresponding transcription. The primary sources are written in various regional historical forms of German, French, Italian, Rhaeto-Romanic, and Latin; the critical apparatuses are in modern German, French, or Italian. The Collection organizes the sources by cantons and then generally subdivides them by areas of jurisdiction, such as towns or bailiwicks. At the moment, the Collection covers 17 of the 26 Swiss cantons to different extents. The edition of the Collection of Swiss Law Sources is an ongoing project and further volumes are in preparation.

The goal of the Collection is to make the sources available to historians of law, law researchers, historians in general, as well as to researchers from other fields and interested laypersons. Extensive back-of-the-book indices ensure that the sources are not only available but also accessible. The Collection is thus an enterprise dedicated to the preservation of Swiss cultural heritage.

The Foundation has recently decided to retrodigitize the volumes published so far to improve their accessibility for research, and to create the infrastructure required for future digital editions. The goal is to make the entire Collection of Swiss Law Sources freely available on the Web. This project can be regarded as a "second phase" of digital cultural heritage preservation, as it does not just digitize the sources but also the associated expert knowledge.



(a) Primary source, a 16th century record of a defamation case (Early New High German; StALU, A1 F1, Sch. 395, Mappe Pfarrei Sargans).

Erstlich hatt Wolfgang Lippuner, landtweibel zů Sarganß, vor einem ersamen gricht zů Sarganß herren Jacoben, caplonen zů Sarganß, anclagt, das er ime am palms abenndt sin meitli altem bruch nach geschickt zů bychten. Inn derselbigen bycht habe er, herr Jacob, zum meitli gredt, es steli ime das höw, die eyer und das kruth unnd derglychen ungschickte frag stuck mit ime, meittli, triben. Weliches aber sich, ob gottwil, nit erfinden werde unnd vermeine derhalben er, herr Jacob, sölle söliches darbringen oder in des meittlis fůßstapffen gestelt werden. Darnebendt er, landtweibel, ouch ettliche wort fallen lassen vor offnem gricht, namlich: «Das üch gotz hie und dört als feldziechen, fasels oder pfaffen schend, wottennd ir mir mine kind zů dieben machen,» etc.

(b) Transcription

Figure 1. Example of a primary source (a) and its transcription (b) in the Collection of Swiss Law Sources (in preparation).

During the work on the retrodigitization we discovered that digital printing data (FrameMaker files) exists for 21 volumes. This is an exciting discovery since it provides a sizeable corpus (about 4 million word forms) of medieval text free from OCR or typing errors.

In the rest of this paper, we will first briefly describe the digitization and then outline the ongoing work towards the use of the Collection as a basis for the creation of language resources.

2 RETRODIGITIZATION

We have scanned all volumes of the Collection of Swiss Law Sources (600 dpi, 1 bit/pixel). We have used optical character recognition

[1] Law Sources Foundation of the Swiss Lawyers Society, Zurich, Switzerland, email: mxp@ssrq-sds-fds.ch

Hasela, Hasele, Haslÿ, Nideren Hasla, Niderhasle, Niderhaslen, Niderhasli, Niderhassla, Nidern Hasel, Nidern Hasela, Nidern Hasele, Nidern Hasla, Nidern Hasle, Nidernhasela, Nidrenhasla, Nidrenhasle, Niederhasli, Niederhaßle, Nydern Hasla, Nÿnnderhasßlenn, nider Hasland, vnder Hasle

Figure 2. Some historical spelling variations for *Niederhasli*.

(OCR) with manual post-correction to convert the tables of contents into digital text. We are using scripts to semi-automatically convert the tables of contents—which differ widely in format—into a normalized XML format. The XML files are then used by the Web viewer to provide navigation facilities to access the facsimiles.

We are not using OCR on the full text of the volumes: Even though the quality of the scanned images is generally high and all volumes have been typeset in roman type (i.e., not in blackletter) from the beginning in 1894, the OCR results are effectively unusable. First, commercial OCR software is unaware of the special characters required for medieval texts (such as å or ů). Second, since there are no lexicons or other linguistic resources for historical language variants (which do not have standardized orthographies), the OCR software has to rely solely on the pattern recognition engine. The OCR results are thus much worse than for modern languages (see [1] for an overview of OCR problems with historical languages). Correcting the OCR results would thus require an inordinate amount of manual correction.

As the Collection contains texts in various languages from a period of about 700 years, full-text retrieval is not useful anyway without special facilities. As an example, figure 2 lists some historical forms of the name of the village today called *Niederhasli*; clearly, a regular search engine is only of very limited use here, and it is better to rely on the manually created back-of-the-book indices, which provide far more information than merely pointers to the occurrences in the text, such as geographic or biographic information in the index of persons and places or grammatical information, such as part-of-speech or gender, in the glossary. See figures 3 and 4 for examples. The main goal of the retrodigitization project is thus to make all volumes available as digital facsimiles.

A

Aachen D 25[36]
Aberli
– *Jakob von Zürich, des Rats* 34[30]
– *Ulrich von Zürich, Obervogt im Neuamt* 44[24], 189[3]
Adlikon (Attlickhon) *Gde. Regensdorf* 8[12]
– *Abgaben* 95[21], 117[18 ff.]
– *Flurnamen:* Breite 111[13], Bůll 110[5], 111[1 f.], Drencke 110[5], Erlin Wisen 110[2], Friessen Studen 111[14], Krumen Acker 110[15], Kruttleren 110[7], Nider Breity 110[36 ff.], Niderbach Tobel 110[10], Nußbom Acker 111[1], Ow 110[19 ff.], Růgacker 110[22], Schwarzen Erd 110[4], Trochen Loo 111[7], Usser Bachtobel 110[15], Winckel Acker 110[24], Witten Strass 110[8]
– *Gemeinde* 39[12], 119[15 ff.]
– *Kilbi* 121[15 ff.]
– *Meier* 5[13], 110[10 ff.], 111[11 ff.]
– *Meierhof* 112[3 ff.], 113[1 ff.], 118[10 ff.]
– *Offnung* 109[2 ff.]
– *Schuldeneinzug* 76[8], 77[5]
– *Weidgang* 111[22 f.]
– *Zehnt* 112[12 ff.]
– *Zehntenscheuer* 39[16 ff.]
Ämperg (Emperg) *Gde. Stadel* 383[18], 386[35]
Affoltern Gde. Zürich 10[32], 84[24]
– ussidelinge 1[26]
Albisrieden Gde. Zürich 232[31 ff.]
Albrecht (Albrächt)
– *von Neerach, Leutnant* 89[22]
– *von Stadel, Hauptmann* 87[22], 96[34], 97[37], 98[10]
– *von Stadel, der Schmied* 329[33]

Figure 3. Extract from an index of persons and places [5, p. 463]

3 EXTRACTING DIGITAL TEXT

As mentioned above, we have the digital printing data in the form of FrameMaker files for 21 of the more recently published volumes. Table 1 lists these volumes, which, in total, amount to 18,879 pages. Based on samples, we estimate the amount of source text to about 4 million word forms. In addition, each volume contains an index of

Arzt 37[37], 329[39]
siehe auch Wundarzt
auffall *siehe Konkurs*
aufheben Stein 9[6], 21[44 f.], 34[24], 251[3 f.]
aufsatz *m.*
1. Auflage 31[16]
2. Entwurf 86[4], 107[31 ff.], 108[2 ff.]
3. Nachstellung, Hinterlist 23[14]
Augenschein 74[34], 140[1 ff.], 192[41], 216[6], 236[19 ff.], 240[1 ff.], 412[3 ff.]
– *Kosten* 180[13 ff.]
Auskauf siehe Erbauskauf
Auslösungsrecht 6[35]
ausrichtung *f. Abfindung um erbrechtliche Ansprüche* 57[40], 62[35], 63[15], 81[10 ff.], 82[27 ff.], 190[41 ff.], 196[21], 200[4 ff.], 235[30 f.], 263[11 ff.]
siehe auch Erbauskauf
Aussiedeling siehe ussideling
Ausstand 445[8]
Baumgarten 84[13], 85[16], 147[19 ff.], 172[20], 248[7 f.], 366[39 f.], 371[15 ff.]
siehe auch büngertlj
beamteter *m. Inhaber eines öffentlichen Amtes* 52[23], 86[17], 88[6], 92[28], 107[17 ff.], 108[2 ff.], 445[5]
siehe auch amtluth
befrygung *f. Privileg, Berechtigung* 66[31 f.]
Begräbnis 91[21 ff.], 94[4]
siehe auch brut und bar
Beherbergung 435[33], 437[28]
Beichte 281[38 f.]
Beischlaf vorzeitiger 97[16 ff.], 101[5 f.]
Beistand in Prozessen 74[44], 75[16 ff.], 171[36], 174[27 ff.], 359[15 f.], 412[20 ff.], 421[35], 422[4]
– *Belohnung* 75[20 f.]
beith *f. Aufschub* 77[25], 198[28]
beketten *v. mit Brettern verkleiden* 114[4]

Figure 4. Extract from a glossary and subject index [5, p. 499]

persons and places and a combined glossary and subject index, all manually created by experts.

We think that this body of text is quite unique. The availability of the full text and indices in digital form enables the traditional target groups of the Collection (historians, historians of law, medievalists, lexicographers, etc.) to access the sources in new ways, such as by spatial browsing; it can also make the sources accessible to new groups of users who are poorly served by the existing indices (e.g., archaeozoologists or interested laypersons).

This body of text is also interesting from the point of view of language technology, as the lack of high-quality digital medieval and early modern texts is generally perceived as a problem for language technology applications on texts from this period. Texts from this period, and German texts in particular, are characterized by a very high level of spelling variation. The addition of further noise in the form of OCR errors and the lack of gold standard texts clearly constitutes an obstacle for the development of NLP methods for historical texts, in particular for tasks such as the identification of spelling variants [2] or the correction of OCR errors [1].

The electronic texts from the Collection represent a sizable corpus of medieval text free from OCR or typing errors in several languages, transcribed, dated, and thoroughly checked by experts, which could be used for the creation of valuable language resources for the processing of medieval and early modern texts.

We have therefore started to convert the texts into XML, so that they can be used for the construction of language resources. While we do not have to cope with OCR errors, the conversion of the texts from a form originally intended only for printing into a form suitable for automatic processing—and eventually language resources—is far from trivial. In the rest of this section, we will give an overview of our approach.

3.1 Converting FrameMaker to XHTML

The FrameMaker files for the volumes mentioned above were produced with FrameMaker versions 3 and 6, beginning in the 1990s. Although newer versions of FrameMaker are to some extent backward compatible, pagination and other aspects of the layout are not necessarily reproduced identically in a later version. Furthermore, the font encoding differs between the Mac, UNIX, and Windows versions of FrameMaker. In the end, the only way to ensure that a document is rendered as originally intended is to open it in the exact version of FrameMaker with which it was produced. We therefore installed FrameMaker 3 and 6 on a Mac OS 9 system and exported all files in MIF (Maker Interchange Format). MIF is an ASCII representation for FrameMaker documents, intended for use by filters or for generation

Table 1. Languages and sizes of the volumes with FrameMaker files. Most volumes also contain sources in Latin. Some volumes consist of several subvolumes.

Volume ID	Canton	Primary Language(s) of the Sources	Pages
SSRQ AG II 9	Aargau	German	735
SSRQ AG II 10	Aargau	German	735
SSRQ AI/AR 1	Appenzell	German	658
SSRQ BE I/13	Berne	German	1143
SSRQ BE II/9	Berne	German	992
SSRQ BE II/10	Berne	German	1191
SSRQ BE II/11	Berne	German	1305
SDS FR I/6	Fribourg	French, German	582
SSRQ GR B II/2	Grisons	German	1403
SSRQ LU I/1	Lucerne	German	592
SSRQ LU I/2	Lucerne	German	481
SSRQ LU I/3	Lucerne	German	731
SSRQ LU II/2	Lucerne	German	2428
SSRQ SG I/2/3	St. Gallen	German	1173
SSRQ SG II/1/1	St. Gallen	German	492
SSRQ SG II/1/2	St. Gallen	German	538
SSRQ SG II/2/1	St. Gallen	German	1184
FDS TI A/1	Ticino	Italian	401
SDS VD B/2	Vaud	Latin, French	622
SDS VD C/1	Vaud	French, German	971
SSRQ ZH II 1	Zurich	German	522
			Total 18,879

by database publishing applications. Syntactically, MIF is relatively easy to parse, since it is documented by Adobe in the *MIF Reference Guide*[2]. The semantics are, however, much harder to determine; for example, the precise inheritance rules for style properties are not documented in the *MIF Reference Guide*. Ultimately, the semantics of MIF statements are defined by the FrameMaker implementation.

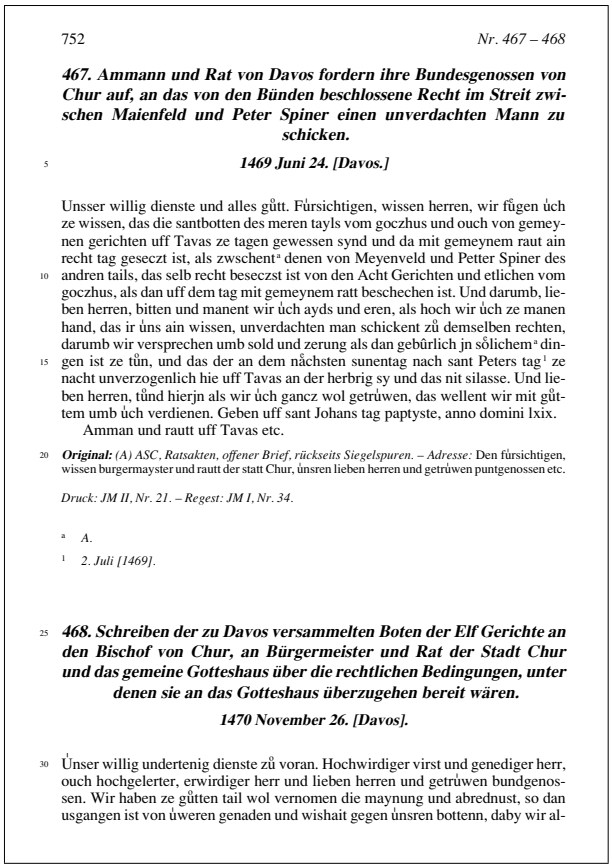
Visually, the printed books and the corresponding FrameMaker documents are consistently marked up. Figure 5 shows an example page: Italics are used for modern text (title, abstract, commentary, apparatus, etc.), while roman is used for source text; the abstract, which also serves as title, is centered and set in a larger font, date and place are centered below; commentary and apparatus are set in a smaller font. In fact, the typography is governed by strict editorial guidelines.

At first sight, the conversion to XML seems to be a simple task, given the consistent use of typographical devices. Unfortunately, in the preparation of the volumes of the Collection, FrameMaker's facilities for structured publishing were not used, but FrameMaker was rather used like a traditional typesetting system. In other words, the typesetter focused solely on the visual appearance: The actual structures in the MIF files are thus much more complex. For example, some named paragraph formats are used, but they are frequently modified *ad hoc* to achieve the desired visual appearance. This means that sometimes a paragraph style is used that specifies roman type and is then locally overridden to use italics or vice versa. Since the distinction between roman and italic is of particular importance to distinguish between source text and commentary, it is not sufficient to look at the specified paragraph format, but the effective font shape must be determined.

Character encoding is a further complicating issue: FrameMaker uses a platform-dependent eight-bit encoding for text, in our case MacRoman.[3] Since the historical texts require special characters not available in MacRoman, special fonts were used that contain special

[2] `http://help.adobe.com/en_US/FrameMaker/8.0/mif_reference.pdf` (accessed 2010-05-17)

[3] Unicode support was only added in 2007 with FrameMaker 8.



752 *Nr. 467 – 468*

467. Ammann und Rat von Davos fordern ihre Bundesgenossen von Chur auf, an das von den Bünden beschlossene Recht im Streit zwischen Maienfeld und Peter Spiner einen unverdachten Mann zu schicken.

1469 Juni 24. [Davos.]

Unsser willig dienste und alles gůtt. Fůrsichtigen, wissen herren, wir fůgen ůch ze wissen, das die santbotten des meren tayls vom goczhus und ouch von gemeynen gerichten uff Tavas ze tagen gewessen synd und da mit gemeynem raut ain recht tag geseczt ist, als zwschent[a] denen von Meyenveld und Petter Spiner des andren tails, das selb recht beseczst ist von den Acht Gerichten und etlichen vom goczhus, als dan uff dem tag mit gemeynem ratt beschechen ist. Und darumb, lieben herren, bitten und manent wir ůch ayds und eren, als hoch wir ůch ze manen hand, das ir ůns ain wissen, unverdachten man schickent zů demselben rechten, darumb wir versprechen umb sold und zerung als dan gebůrlich jn sölichem[a] dingen ist ze tůn, und das der an dem nåchsten sunentag nach sant Peters tag[1] ze nacht unverzogenlich hie uff Tavas an der herbrig sy und das nit silasse. Und lieben herren, tůnd hierjn als wir ůch gancz wol getrůwen, das wellent wir mit gůttem umb ůch verdienen. Geben uff sant Johans tag paptyste, anno domini lxix.

Amman und rautt uff Tavas etc.

Original: *(A) ASC, Ratsakten, offener Brief, rückseits Siegelspuren. – Adresse:* Den fůrsichtigen, wissen burgermayster und rautt der statt Chur, ůnsren lieben herren und getrůwen puntgenossen etc.

Druck: JM II, Nr. 21. – Regest: JM I, Nr. 34.

[a] *A.*

[1] *2. Juli [1469].*

468. Schreiben der zu Davos versammelten Boten der Elf Gerichte an den Bischof von Chur, an Bürgermeister und Rat der Stadt Chur und das gemeine Gotteshaus über die rechtlichen Bedingungen, unter denen sie an das Gotteshaus überzugehen bereit wären.

1470 November 26. [Davos].

Ůnser willig undertenig dienste zů voran. Hochwirdiger virst und genediger herr, ouch hochgelerter, erwirdiger herr und lieben herren und getrůwen bundgenossen. Wir haben ze gůtten tail wol vernomen die maynung und abrednust, so dan usgangen ist von ůweren genaden und wishait gegen ůnsren bottenn, daby wir al-

Figure 5. Example page from an edited volume, illustrating layout and typographic devices used to distinguish between source texts and commentary. [6, p. 753]

glyphs (e.g., ů) in the positions normally used for other characters (e.g., ø). This means that one cannot simply convert the encoding of the whole text in one go, but the meaning of a byte always depends on the current font, which again depends on the paragraph and character formats in effect.

Since the structure of the documents and the encoding are crucial, these and other issues preclude simple approaches that do not analyze the MIF file in depth but rely on the consistent use of named paragraph formats (e.g., [7]). Starting with version 5, FrameMaker allows to export documents as XML or HTML; however, we had to rule out this option, since it requires to use a version different from the one the documents were created with, which creates compatibility problems. What is more, while FrameMaker 7 was able to export the prefatory matter of the tested volumes as XML, it was unable to export the main part.

We therefore decided to develop our own converter. Our goal is to eventually produce a corpus marked up according to TEI [8]. However, due to the complexity of the MIF files, a one-step upconversion is not feasible. We thus decided to first convert MIF into XHTML with an associated CSS stylesheet that captures as much information from the original MIF as possible and which then could be used for further processing.

The resulting program, *mif2html*, uses the FrameMaker::MifTree[4] Perl module by Roel van der Steen to build a full parse tree of MIF files. It then processes all paragraph and character format specifications, taking into account MIF's formatting model and inheritance model, to produce equivalent CSS styles. CSS and MIF's format language differ significantly in the way they are processed: While CSS is declarative, MIF is procedural, i.e., a format can be changed at any place in the document and the modified format is in effect from that point on.

mif2html also converts the character encoding to Unicode in dependence on the font. The pagination and the line numbering information are also transferred to XHTML, since they are required to refer from the indices to the occurrences in the text, so that the indices can, for example, augmented with hyperlinks. The original line breaks can be either converted—the XHTML document then corresponds line by line to the printed book—or the lines can be joined and hyphenation removed. The resulting XHTML document is valid and, when displayed in a Web browser, closely corresponds to the printed book. *mif2html* currently only handles the MIF commands used in our collection, but since actually parses MIF files and interprets them according to the specification, it could also be applied to other MIF files.

3.2 Further processing and use

Since the XHTML output closely mirrors the original MIF file, it contains a large amount of markup that is superfluous if one is interested in extracting the actual text, whether it is source text, commentary, or index entries. In the next step, the output of *mif2html* is therefore converted into a more concise form. We still do not try to infer any semantics at this point. The following is a paragraph from an index of persons and places:

```
<p class="register" id="tf10p28">
<span id="tf10p28f1">Aachen</span>
<span id="tf10p28f2"> </span>
<span id="tf10p28f3">D</span>
<span id="tf10p28f4"> </span>
<span id="tf10p28f5">25</span>
<span id="tf10p28f6">36</span></p>
```

This input results in the following output:

```
<dfn><i>Aachen</i></dfn> <i>D</i> 25<sup>36</sup>
```

In order to make this simplification, a cleanup program reads the CSS stylesheet and looks up the relevant properties of each paragraph and each span to determine whether the effective style is italic, letterspaced, superscript, etc. It also suppresses irrelevant markup, such as that of whitespace between two other elements (as in the example above).

Once the XHTML has been simplified, it becomes feasible to determine the function of elements and to add semantic information or to extract certain elements. It can thus be used for linguistic exploration and for the construction of language resources. For example, we are using context-free grammars and a recursive-descent parser[5] to identify and extract the elements of glossary entries (such as headword, part of speech, glosses, variant spellings, page and line references, etc.). We are currently working—among other things—on linking the headwords of glossary entries to the occurrences in the text to build up a dictionary of attested forms and spelling variants and to experiment with suitable retrieval methods; Pilz et al. [3] have evaluated a number of approaches, but it is not yet clear how they will perform on the Collection of Swiss Law Sources. We have also used the index of persons and places for experiments in spatial browsing, i.e., for creating maps that allows to browse and access the texts from a geographic perspective [4].

[4] `http://search.cpan.org/~rst/FrameMaker-MifTree/`

[5] Using the Parse::RecDescent Perl module by Damian Conway, `http://search.cpan.org/~dconway/Parse-RecDescent/`

4 CONCLUSION

In this paper, we have briefly presented the Collection of Swiss Law Sources and the current retrodigitization project. We then have focused on the portion of the Collection for which FrameMaker files are available (18,879 pages). Since very little high-quality medieval and early modern text is available in electronic form, we have argued that this body of text constitutes a valuable asset for the development of language resources and NLP techniques, in particular for medieval and early modern German, and, to a smaller extent, for French and Italian. In addition to the source texts, the collection is complemented by indices of persons and places—especially interesting for work on named entity recognition—and glossaries, which include grammatical information and spelling variants. However, to be usable for these purposes, the texts must first be extracted from the FrameMaker files and brought into a usable form. While the conversion process does not involve language technology in the narrow sense, it is a prerequisite for future work in the area of language technology applications for cultural heritage, social sciences, and humanities.

We have described the work performed up to now to make the texts from this subset of the Collection usable for purposes of language technology and NLP. At the time of this writing, we have a set of tools to convert FrameMaker files to XHTML and to prepare the XHTML for further processing. Our tools differ from *ad hoc* approaches for converting FrameMaker data in that they parse and interpret the MIF files according to the specification, thus being more general and better suited for reuse in other projects. We have also carried out some preliminary work towards the extraction of linguistic knowledge. This is ongoing work; we intend, on the one hand, to complete the construction of language resources from the texts and indices and, on the other hand, to use these resources to improve the access to both this subset of the Collection and to the larger part for which no full text is available.

ACKNOWLEDGEMENTS

I would like to thank Pascale Sutter for collating the example transcription at short notice and for her historical expertise in general. The work described in this paper was supported by the Swiss National Science Foundation.

REFERENCES

[1] A. Gotscharek, U. Reffle, C. Ringlstetter, and K. U. Schulz, 'On lexical resources for digitization of historical documents', in *DocEng '09: Proceedings of the 9th ACM symposium on Document engineering*, pp. 193–200, New York, NY, USA, (2009). ACM.

[2] T. Pilz, A. Ernst-Gerlach, S. Kempken, P. Rayson, and D. Archer, 'The identification of spelling variants in English and German historical texts: Manual or automatic?', *Literary and Linguistic Computing*, **23**(1), 65–72, (April 2008).

[3] T. Pilz, W. Luther, and U. Ammon, 'Retrieval of spelling variants in nonstandard texts – automated support and visualization', *SKY Journal of Linguistics*, **21**, 155–200, (2008).
[4] M. Piotrowski, 'Leveraging back-of-the-book indices to enable spatial browsing of a historical document collection', in *Proceedings of the 6th Workshop on Geographic Information Retrieval (GIR'10)*, eds., R. Purves, P. Clough, and C. Jones, pp. A17/1–2, New York, NY, USA, (February 2010). ACM.
[5] *Das Neuamt*, ed., Rechtsquellenstiftung des Schweizerischen Juristenverbandes, volume SSRQ ZH I/1 (I. Abteilung: Die Rechtsquellen des Kantons Zürich, Neue Folge: Zweiter Teil: Rechte der Landschaft, Band 1) of *Sammlung Schweizerischer Rechtsquellen*, Sauerländer, Aarau, Switzerland, 1996. Prepared by Thomas Weibel.
[6] *Landesherrschaft und Bundesrecht*, ed., Rechtsquellenstiftung des Schweizerischen Juristenverbandes, volume SSRQ GR B II/2 (XV. Abteilung: Die Rechtsquellen des Kantons Graubünden, B. Die Statuten der Gerichtsgemeinden, Zweiter Teil: Der Zehngerichtenbund, Band 2) of *Sammlung Schweizerischer Rechtsquellen*, Schwabe, Basel, Switzerland, 2008. Prepared by Elisabeth Meyer-Marthaler and Martin Salzmann, indices by Evelyn Ingold.
[7] B. Rousseau and M. Ruggier, 'Writing documents for paper and WWW: a strategy based on FrameMaker and WebMaker', in *Selected papers of the first conference on World-Wide Web*, ed., R. Cailliau, pp. 205–214, Amsterdam, The Netherlands, (1994). Elsevier.
[8] C. Wittern, A. Ciula, and C. Tuohy, 'The making of TEI P5', *Literary and Linguistic Computing*, **24**(3), 281–296, (2009).