Adaptation of Language Resources and Tools for Processing Cultural Heritage Objects

Workshop Programme

09:00 – 09:30 – Introduction by Workshop Organisers

09:30 – 10:30 Invited talk: Stefanie Diepper. *Automatic methods for historical language data: studies on rule-based normalization, part-of-speech and morphological tagging*

10:30 – 11:00 Coffee break

11:00 – 11:20 Mikhail Gronas, Anna Rumshisky, Aleksandar Gabrovski, Samuel Kovaka and Hongyu Chen. *Tracking the history of knowledge using historical editions of Encyclopedia Britannica*

11:20 – 11:40 Jirka Hana, Boris Lehečka, Anna Feldman, Alena Černá and Karel Oliva. *Building a corpus of Old Czech*

11:40 – 12:00 Agnieszka Mykowiecka, Piotr Rychlik and Jakub Waszczuk. *Building an electronic dictionary of Old Polish on the base of the paper resource*

12:00 – 12:20 Serge Heiden and Alexei Lavrentiev. *The TXM portal software giving access to Old French manuscripts online*

12:20 – 12:40 Lauma Pretkalnina, Peteris Paikens, Normunds Gruzitis, Laura Rituma and Andrejs Spektors. *Making historical Latvian texts more intelligible to contemporary readers*

12:40 – 13:00 Baldev Ram Khandoliyan, Rajneesh Kumar Pandey, Archana Tiwari and Girish Nath Jha. *Text encoding and search for Āyurvedic texts: an interconnected lexical database*

13:00 – 13:20 Tobias Sippel and Jan-Torsten Milde. *A multitouch enabled annotation editor for digitized historical documents*

13:20 – 13:40 Dolf Trieschnigg, Djoerd Hiemstra, Mariët Theune, Franciska de Jong and Theo Meder. *An exploration of language identification techniques for the Dutch Folktale Database*

13:40 – 13:45 Closing session

Editors

Petya Osenova	St. Kl. Ohridski University of Sofia and IICT, Bulgarian Academy of Sciences, Bulgaria
Stelios Piperidis	Institute for Language and Speech Processing, Athens, Greece
Milena Slavcheva	IICT, Bulgarian Academy of Sciences, Bulgaria
Cristina Vertan	University of Hamburg, Germany

Workshop Organisers

Petya Osenova	St. Kl. Ohridski University of Sofia and IICT, Bulgarian Academy of Sciences, Bulgaria
Stelios Piperidis	Institute for Language and Speech Processing, Athens, Greece
Milena Slavcheva	IICT, Bulgarian Academy of Sciences, Bulgaria
Cristina Vertan	University of Hamburg, Germany

Workshop Programme Committee

David Baumann	School of Computer Science, Carnegie Mellon, USA
Walter Daelemans	University of Antwerp, Belgium
Günther Görz	University Erlangen, Germany
Walther v. Hahn	University of Hamburg, Germany
Piroska Lendvai	Hungarian Academy of Sciences, Hungary
Anke Lüdeling	Humboldt University, Berlin, Germany
Gábor Prószéky	MorphoLogic, Hungary
Laurent Romary	LORIA-INRIA, Nancy, France
Éric Laporte	Université Paris-Est Marne-la-Vallée, France
Kiril Simov	IICT, Bulgarian Academy of Sciences, Bulgaria
Manfred Thaler	Cologne University, Germany
Tamás Váradi	Hungarian Academy of Sciences, Hungary
Martin Wynne	University of Oxford, U.K.
Kalliopi Zervanou	University of Tilburg, The Netherlands

Table of Contents

Author Index

Preface

Recently, the collaboration between the NLP community and the specialists in various areas of the Humanities has become more efficient and fruitful due to the common aim of exploring and preserving cultural heritage data. It is worth mentioning the efforts made during the digitisation campaigns in the last years and within a series of initiatives in the Digital Humanities, especially in making Old Manuscripts available in the form of Digital Libraries.
Most parts of these libraries are made available not only to researchers in a certain Humanities domain (such as, classical philologists, historians, historical linguists), but also to common users. This fact has posited new requirements to the functionalities offered by the Digital Libraries, and thus imposed the usage of methods from Language Technology for content analysis and content presentation in a form understandable to the end user.

There are several challenges related to the above mentioned issues:

- Lack of adequate training material for real-size applications: although the Digital Libraries usually cover a large number of documents, it is difficult to collect a statistically significant corpus for a period of time in which the language remained unchanged.
- In most cases, the historical variants of the language lack firmly established syntactic or morphological structures and that makes the definition of a robust set of rules extremely difficult. Historical texts often constitute a mixture of several languages including Latin, Old Greek, Slavonic, etc.
- Historical texts contain a great number of abbreviations, which follow different models.
- The conception of the world is somewhat different from ours (that is, different thinking about the Earth, different views in medicine, astronomy, etc.), which makes it more difficult to build the necessary knowledge bases.

Having in mind the number of contemporary languages and their historical variants, it is practically impossible to develop brand new language resources and tools for processing older texts.
Therefore, the real challenge is to adapt existing language resources and tools, as well as to provide (where necessary) training material in the form of corpora or lexicons for a certain period of time in history.

The current workshop tries to address those issues. The proceedings contains eight papers dealing with historical variants of languages such as French, Dutch, German, Czech, Polish, Latvian, Sanscrit. The paper topics range from the creation of language resources and their intelligent representation for non-specialist users to the development of automatic tools for processing historical language variants.

We would like to thank all contributors, our invited speaker, and especially the members of the programme committee who completed the review process in extremely short time.

The Organisers

Automatic Methods for Historical Language Data: Studies on Rule-Based Normalization, Part-of-Speech and Morphological Tagging

Stefanie Diepper

Analysis of historical languages differs from that of modern languages in two important points. First, there are no agreed-upon, standardized writing conventions. Instead, characters and symbols used by the writer of some manuscript in parts reflect impacts as different as spatial constraints (parchment is expensive and, hence, use of abbreviations seems favorable) or dialect influences (the dialect spoken by the author of the text, or the writer's dialect, who writes up or copies the text, or even the dialect spoken by the expected readership). This often leads to inconsistent spellings, even within one text written up by one writer. Second, resources of historical languages are scarce and often not very voluminous.

These features - variance in the data and lack of large resources - challenge many statistical analysis tools, whose quality usually depend on the availability of large training samples. A common way to tackle these problems is by normalizing historical spellings, by mapping them to modern wordforms or some virtual historical standardized forms.

In the talk, I will present an unsupervised, rule-based approach to wordform normalization. Rules are specified in the form of context-aware rewrite rules that apply to sequences of characters. The rules are derived from two aligned versions of the Luther bible. I will also present results from a set of tagging experiments. In these experiments, a state-of-the-art tagger is applied to original and normalized wordforms, to assign part-of-speech and morphological tags. The data used in this research are texts from Middle and Early New High German.

Tracking the History of Knowledge Using Historical Editions of Encyclopedia Britannica

M. Gronas[∗‡], **A. Rumshisky**[†], **A. Gabrovski**[∗], **S. Kovaka**[∗], **H. Chen**[∗]

MIT, Cambridge, MA, USA[†]
Dartmouth College, Hanover, NH, USA[∗]
Center for Media and Society, NES, Moscow[‡]

Abstract

Despite the wealth of newly available digital materials, the scope of text-based investigations has mostly been limited to either synchronous or short-term historical analysis. In this paper, we report on the first stage of the project that focuses on tracking long-range historical change, specifically, on the history of ideas and concepts. The project's aim is to map out the history of representation of knowledge in Europe over last three centuries using as a proxy the history of changes in historical editions of Encyclopedia Britannica. We describe a series of corpus-analytical tasks necessary for building the analytical and comparative tools for historical analysis using scanned noisy text. In this first stage of the project, we focus specifically on the tools for tracking and visualizing the relative importance of people, interconnections between them, and the rise and fall of their reputations.

1. Introduction

Humanities, especially historical disciplines, such as literary history, or political history, or history of science, study changes and evolutions in their respective fields; all such disciplines greatly benefit from tools that map, track down, and measure historical changes and trends within and across their respective fields. But the data in question - e.g. literary and philosophical schools coming and going, scientific theories thriving and falling in disrespect, reputations of authors, musicians, politicians either surviving or falling apart, new gadgets getting invented, new beliefs adopted, fashions, movements, zeitgeists spreading and disappearing, etc. - stems from the pre-digital ages and seems too noisy for digital methods.

In this work, we use historical editions of Encyclopedia Britannica as an (admittedly imperfect) proxy for the history of knowledge in modern Europe. Since its glorious birth in the Age of Enlightenment, the modern encyclopedia has indeed served as the standard representation of human knowledge. The ambition to give both the fullest and most up-to-date account of the state of knowledge gradually transformed the encyclopedia into a never ending collective enterprise, continuously built and rebuilt over centuries by the generations of scholars and editors, our civilization's answer to the Gothic cathedral of the Middle Ages. As the new ideas arise and old ones disappear, fields and domains gain in importance or shrink, encyclopedias respond by mirroring the changes in new editions and updates. Since encyclopedias reflect consensus of scholarly opinions and, by definition, aspire to universality and balance, a single edition may serve as – although obviously imperfect, but best available – representation a synchronic layer of contemporary knowledge, a panoramic snapshot of the state of knowledge. Then the succession of such snapshots, a moving picture of multiple successive editions of the same encyclopedia viewed in a historical perspective, may serve as the best available approximation of the data set on the history of knowledge.

In this paper, we discuss the analytical methods for the historical analysis of conceptual domains. We apply natural language processing and network analysis techniques to the corpus of several historical editions of encyclopedia Britannica; we use the current edition Britannica and to Wikipedia to supplement the analyses. In the present study, we have limited our dataset to the articles about people thus focusing on the social dimension of the history of knowledge. Historical editions in our corpus are OCR scans, and therefore contain very noisy data. In order to use these texts for comparative analysis, we had to perform a series of corpus-analytical tasks, which we describe in this paper. These include:

1. Splitting the text into articles and identifying article titles
2. Identifying articles about people
3. Matching the articles across editions
4. identifying explicit references to other articles.
5. Identifying article categories and matching them across editions.

Once these tasks are performed, the texts can be analyzed for cross-edition changes.

The remainder of this paper is organized as follows. In Section 2., we outline our general approach to formalization of the historical analysis task. In Section 3., we describe the data set we used in our analyses. In Section 4., we describe in some detail the text processing tasks that are required in order to build the diachronic database. We then describe cross-edition analysis methods applied to the derived marked-up text and present the interface for browsing concepts across editions.

2. Formal Description

Our analytical methods are based upon the following generalized model of what a traditional Encyclopedia does:

(1) Selection:
A finite set of areas, concepts, and personalities worthy of inclusion are selected out of the infinity of potential subjects;

(2) Ranking:
The importance of each subject is ranked by the volume assigned to each entry, with more important subject taking up more space;

(3) Interpretation:
Each subject is described and interpreted, and furthermore, placed in relation to other subjects (e.g. through comparison classification, hierarchical subdivisions etc.), thus defining its relative position on the map of contemporary knowledge.

These three steps selection, ranking, and interpretation of the concept and its relative position – inform the changes from one edition to another. Each aspect of concept representation may undergo changes, in particular:

(1) The list of selected subjects grows and changes, picking up new theories, inventions, persons etc, and shedding the ones which are no longer deemed worthy of inclusion;

(2) The volume devoted to a subject also fluctuates mirroring changes in its perceived importance, e.g. the relative space allotted to Shakespeare in Britannica has been steadily growing through 17 and 18 centuries reflecting his growing reputation as the center of the Western Literary Canon, whereas space devoted to, e.g. natural philosophy has been shrinking.

(3) The relations between concepts also evolve through continuous changes in classifications, hierarchies and patterns of associations: in successive editions, a concept may be reclassified, placed in another domain or sub-domain and related to a different set of concepts.

To capture these changes, for each subject (encyclopedic entry on a person) in each of the editions we determine the following relevant factors:

- *Inclusion*, corresponding to presence or absence of an entry in a given edition
- *Size*, corresponding to the percent of the total volume of the edition dedicated to that entry;
- *Centrality*, corresponding to the concept rank, based on several parameters, including the number of incoming references to a given entry, the number of mentions of the subject in other entries, etc.
- *Position*, corresponding to the relations between subjects/domains as represented in concept co-occurrence patterns, clique membership in the graph induced from the set of encyclopedic entries, etc.

The subjects (i.e., in the pilot dataset, persons) are organized into domains, constellations of people belonging to the same field. The historical changes happen both within and across the domains: subjects get included or excluded (factor Inclusion), they grow or shrink in importance (factors size/centrality), they change their position relative to other subjects constituting the domain (factor position). We describe particular tools and methods we develop to track and describe these factors in Section 4..

3. Data set

We used three scanned and OCR'ed out-of-copyright editions of Britannica: Editions 3 and 9 by GoogleBooks, and Edition 11 available from jrank.org. Encyclopedia Britannica granted us research rights to use the electronic text of Britannica's current (15th) edition in XML format. We also used Wikipedia, which effectively provides an (extensive) update to the 15th edition, since the original articles from Britannica's older edition often served as the initial version for Wikipedia entries. These editions gave us the total of 5 points of comparison:

1. Edition 3 (1788–1797)
2. Edition 9 (1875–1889)
3. Edition 11 (1910–1911)
4. Edition 15 (Current EB; 1985–Present)
5. Wikipedia (Present)

Our choice of the specific historical editions was motivated by the fact that some of they represent distinct changes in the state of Encyclopedia Britannica. Edition 3 represents the initial period in the history of Encyclopedia Britannica when it was just being established as an authoritative source. Edition 9, hailed as a Scholar's edition, represented a considerable reworking of the previous editions, with multiple respected authorities in different fields of knowledge contributing articles. Edition 11 again represented a change in the state of knowledge; it was a complete reworking of the Encyclopedia, and remained an authoritative source for several decades.

3.1. Restricting Subject Set

Ideally, the comparative analysis should be conducted using all the subjects for which articles are present in the Encyclopedia, i.e. concepts corresponding to the article titles. This set of concepts should be correlated with and supplemented by the concepts extracted from the text of the articles.

However, even using only the concepts from article titles we encounter a considerable amount of noise. In the present work, we restricted our data to articles on people only for the following reasons. Our analysis is based upon matching of the articles between editions (locating articles on the same topic across edition). Also, in order to conduct the full analysis of conceptual relations and relative importance of different concepts, we detect mentions of different subjects from within other articles, effectively creating a hypertext structure. At present stage, this is exceedingly complicated when dealing with most concepts expressed by common nouns, because of (a) differences in taxonomies: same concept can be part of an article in an earlier edition and has its own article in a later edition b) polysemy: same word can serve as a head word of an article (e.g. nature) and be used in a different meaning (e.g. in proposition by nature). A sturdy disambiguation in such cases is highly problematic. However, proper names (e.g. persons), while still constituting a hugely important domain of knowledge, are less affected by these limitations because : a) persons are usually classified as such; b) namesakes are relatively easy to disambiguate.

4. Tasks

4.1. Cleaning up the data

Historical editions in our corpus are OCR scans, and therefore contain very noisy data. A large proportion of all words are mis-scanned, with text segments from different articles interspersed. The initial task was therefore to (1) split each of the historical editions into separate articles and (2) identify titles for each article. Edition 11 was obtained from the jrank.org website, where it was pre-split into articles. Despite being collectively edited it does contain a significant amount of errors. Edition 11 is also in progress of being manually corrected as part of Project Gutenberg. We replaced the first 13 volumes of text with the manually corrected volumes.

For Editions 3 and 9, we opted not to build a classifier for this auxiliary task. Rather, splitting the text into articles and title identification was performed using a set of simple formatting heuristics, such as looking for uppercase strings at the beginning of paragraphs preceded by blank lines; eliminating mis-scanned tables by identifying text segments with comparatively small average line length, etc. This was complemented by an *alphabetic ordering check* on title candidates. The latter entails checking the alphabetic ordering of the set of proposed titles to remove some of the false positives that break the ordering.

Checking that the next article is in the correct position alphabetically is not sufficient, since one mis-scanned title could cause all subsequent articles to be marked as bad. For example, if there were three articles in a row titled “AARDVARK”, “ABLE”, and “ACE”, they would be all be marked correct because they are in alphabetical order. If, however, “ABLE” was mis-scanned as “AELE” it would still be correct because it still comes after “AARDVARK”, but then “ACE” would be incorrect because it should come before “AELE”. To remedy this, we first find all possible articles by just searching for upper case letters, then assign each article a score based on how many articles before it are in fact alphabetically before, and how many articles after it are in fact alphabetically after, using a threshold to filter out false positives.

Under this setup, a large group of bad titles could cause many nearby articles to get a low score. We therefore first run title extraction a low threshold to get rid most large groups of bad titles, followed by several runs with higher cutoffs to weed out the remaining stragglers. Alphabetic ordering check also had to take into account miscellaneous issues such as the fact in that in some of the older editions letters U and V, as well as J and I, were used interchangeably.

We conducted some accuracy testing by manually checking the accuracy of the split-and-extraction algorithm on a subset of the extracted articles. The following estimates for error rates were obtained:

Edition 3 (GoogleBooks) 19.0% error rate
Edition 9 (GoogleBooks) 10.1% error rate
Edition 11 (Jrank) 14.7% error rate

The OCR’ed editions are quite noisy, and we conducted some quantitative investigations of correctness with a modified spell-checker tool which relies on the current edition of Britannica as well as on Wikipedia for lexical information. For the editions obtained from Google Books, considering only the tokens consisting of alphabetic characters with punctuation, the percentage of misspelled words varied across volumes as follows:

- 7.1–9.6 % in Edition 3
- 5.3-7.9% in Edition 9

4.2. Processing graph structure from individual editions

We have investigated several approaches to constructing article graphs. For each edition, two main types of graphs are currently constructed:

1. Distance graphs, using co-occurrence statistics
2. Explicit reference graphs

We have developed software that allows experimentation with different weighting techniques to tune ranking and clustering methods, with preliminary results available for PageRank and Markov Clustering on both types of graphs for each edition. We ran PageRank on both types of graphs, producing importance ranks for individual articles within each edition.

For each edition, we also ran Markov Clustering on the explicit reference graph in order to partition the articles into distinct clusters. Inflation rate, a factor affecting the segmentation of clusters, was an important part of the the algorithm. This parameter was determined by observing the distance between two different clusterings (the number of node changes required to convert one clustering into another) and cluster tightness. Cluster tightness was determined by the product of the Jaccard similarity to each article’s wikipedia categories and the cluster size. By averaging this score over all clusters for each inflation value, we could objectively select appropriate Markov Clustering parameters for every edition.

The purpose of clustering all articles from every edition is twofold. First, it can serve as a comparison method between articles from within an edition. Second, and more importantly, clusters can track when certain ideas are no longer associated, at least algorithmically, with what it was associated with before.

4.3. Normalization of articles across different editions

We have done cross-edition normalization using Wikipedia categories and the metadata from the current Britannica editions. The cross-edition mapping approach we have been investigating involves the mapping of different categories from the Wikipedia and Current EB to article sets across historical editions. This involves normalization and mapping of article titles to enables the category mapping. Current approaches we are taking involve distributional ranking of article similarity with differential weighting of different article segments, as well as incorporating weighted use of Wikipedia suggestions and targeted Bing searches. We give more detail on this task in the following section.

5. Cross-edition article matching

We used TF*IDF to obtain weighted word vector representations of each article. Since the beginning of the article, i.e. the title and the introduction, usually contain a concise version of the most important information laid out in the rest of the article we over-weighed the beginning of each article, in particular giving more weight to strong identificators such as personal names, dates, names of the professions.

For each edition obtained, every article is first matched to the Wikipedia article on the same topic. The articles matching the same Wikipedia article are then matched across editions as follows.

Step 1. Finding candidates for matching

First we use a list of all article titles in Wikipedia which is sorted alphabetically. An insertion index is obtained for the spot where the title of the article in question can be inserted while preserving the sorted order of the list. Then the surrounding k articles around the insertion index are added to our candidate list (we used $k = 6$ in the experiments below). We then query Wikipedia and Bing for each title of an article and add the top 5 results of each query to our candidate list. The final candidate list of Wikipedia articles is compiled by resolving redirects, removing missing articles and adding candidates from disambiguation pages.

Step 2. Candidate Comparison

Each candidate Wikipedia article is compared to the article we are trying to match using a cosine similarity measure computed for the corresponding weighted TF*IDF word vectors.

Step 3. Detecting articles about people

We apply Wikipedia's categories to filter out non-person articles. Wikipedia articles about people are often assigned categories specifying birth or death year (e.g. the article about Johann Sebastian Bach belongs to the categories 1685 births and 1750 deaths). If an article is not assigned a birth-year or death-year category, the original article from Encyclopedia Britannica is filtered out.

Parameter tuning

The above algorithm uses the following parameters:

1. number of words overweighed in the beginning of article ($first\_word\_lim$);
2. number of words in the beginning considered to the title, usu. names and years ($title\_word\_lim$);
3. weight factor for the title words ($title\_word\_ow$);
4. weight factor for for years found in the beginning of the article ($year\_ow$;
5. weight factor for professions/occupations (e.g. author, poet, tsar) found in the beginning of the article (occ_ow);
6. whether a given word would only be overweighed once; e.g. an occupation might be mentioned multiple times ($only\_once\_ow$)
7. number of words from the article to query Bing and Wikipedia with, along with the title (num_words_q)

We tuned the parameters sing a manually annotated set of 100 articles for each edition. A randomly selected set of articles from Britannica obtained from the initial run of the algorithm was manually matched against the Wikipedia articles and used for parameter tuning. The error rate was then estimated on another 100 of manually matched articles. Table 5. summarizes optimal parameter settings for Edition 9.

The estimates we obtained suggest 75% accuracy, with the error rates for the parameters specified above as follows:

1. incorrect matches - 14.61%
2. non-person articles - 5.26%
3. person articles filtered out - 5.26%

Assigning categories across editions

We used Wikipedia categories to generalize across editions, the rationale is that such a categorization will allow us to track the development of topics, as well as specific articles.

Wikipedia's categories, while benefiting from the wisdom of the crowds, also inherit problems associated with it. A lot of categories are ad-hoc and not every article in Wikipedia has been assigned all categories that it should conceptually have. An alternative is to use Encyclopedia Britannica's internal tagging system used in the current electronic edition. Each article in our corpus is matched to its edition 15 counterpart (using our Wikipedia matching as crutch) and get its categories.

6. Browsing Interface and Query Tool

One of the results of our project is an educational online tool for tracking and mapping the social dimension of the history of knowledge, or put simply, a history of reputations. The back-end of the tool is a database that contains all articles about people in different Encyclopedia editions (Britannica 3 ,9 , 11, 15, and Wikipedia). Articles on the same person are matched across editions to compile a master list of people. Each subject is characterized by measures of importance and centrality in respective editions, by the network of co-occurring subjects ("neighbours") , and by the list of categories accompanying this subject in Wikipedia.

The user first picks the domain of interest. The domain is effectively, a Wikipedia category, or a cross-section of Wikipedia categories or lists (e.g. French 19th century composers, chemists, members of the romantic movement, etc). Once the domain is picked, the system produces the snapshots of the domain for each historical edition of the Britannica and for Wikipedia. The snapshots are maps containing all participants of the domain as presented by the respective editions. The more important the subject the larger his or her node on the map; the more central the subject the more central the node; the more frequently two subjects co-occur, the closer they are on the map. The snapshots are then displayed in succession so as to create a movie-like dynamic representation of how domains (actors, their importance, relations between them) changed over the course of last three centuries.

$year\_ow$	occ_ow	$title\_words\_ow$	$once\_only\_ow$	$first\_words\_lim$	$title\_word\_limit$	num_words_q
12	12	8	True	150	4	5

Table 1: Parameter values for Edition 9

6.1. "Gravebook"

One of the products of our research is a novel education tool that highlights and makes more accessible for students the social and intellectual networks of the past. This tool allows one to investigate social and intellectual connections of persons featured in the Current Edition of Britannica and Wikipedia, and reconstruct the underlying social graph, thus creating, effectively, a facebook for the past, or as we term it, Gravebook, an entertaining interface for studying history of human connections. The interface is built as follows. We first determine all entries about people; then all links to other people-articles contained in these entries. For each such link we calculate if the linked person was born after or before the subject of the article; if the life spans of the two overlap, then this connection is considered to be a likely personal acquaintance. Based on this analysis, we will create mock up facebook pages for all persons mentioned in encyclopedias, with linked friends accounts, likes etc. An alternative interface is a 3D visualization of social networks of the past, based on the spring-box algorithms.

7. Related Work

Transmission and diffusion of information, as well as the visualization of connections between different concepts and areas of knowledge, has attracted the attention of scholars is many different research areas. Most of this work has been done in the context of *analysis of social networks*, ranging from product marketing applications (Brown and Reinegen, 1987; Mahajan et al., 1990; Domingos and Richardson, 2001) to the spread of innovation and best practices in medicine and other areas (Coleman et al., 1966; Rogers, 1979) to strategy adoption in game-theoretic settings (Young, 1998; Morris, 2000). Dissemination of ideas in science and the impact of particular works on a given domain has also been the subject of study in scientometrics and bibliometrics, in a long tradition dating back to the works of Rice (Rice, 1965) and Garfield (Garfield, 1955; Garfield, 1972) who pioneered the use of academic *citation patterns* to determine the *impact factor* of the work of particular scientists or scientific journals as publishing entities. The work in scientometrics produced some relevant analytical and visualization tools, e.g. for tracking the life cycle and impact of scientific paradigms (Boyack et al., 2005). An example is the Map of Scientific Paradigms (Boyack et al., 2005) which tracks the relationships between different areas of research in sciences and social sciences by looking at inter-citation and co-citation between scientific journals. Garfield's work on citation patterns had also given rise to the *link structure analysis* algorithms that have been used more recently in the analysis of web graph structure and the ranking of web pages in search applications, including Google's *PageRank* (Page et al., 1998), HITS (aka *hubs and authorities* algorithms, or hypertext-induced topic search) (Kleinberg, 1999), and others. More recently, similar methods have been applied to the tracking emergent trends and topics in dynamic data sets, such as email correspondence, news articles, and the blogosphere. For example, Kleinberg (2003)) tracked intensity of topics in email and news articles. Leskovec et al. (2009)) developed the "Memetracker", a data analysis and visualization tool which tracked the proliferation in the news stories and blog posts of catch phrases mentioned by the candidates in the 2008 U.S. presidential election. Gloor et al. (2008)) measured concept's relative importance by looking at the number of paths in the network that go through that concept (*betweenness centrality*), applying this method to the documents retrieved from the web for representative phrases in a particular domain, such as names of politicians, brands, etc.

In the analysis of groups in online communities, the focus has been both on (1) identifying subcomminities within a particular network and tracking their development over time (Kumar et al., 2005; Chi et al., 2007), and (2) determining the influential nodes that contribute more to the diffusion of information within the network (Gruhl et al., 2004; Adar and Adamic, 2005; Nakajima et al., 2005; Kimura et al., 2007). The latter has been modeled, for example, on the theory of *propagation of infectious diseases* in epidemiology. For example, Adar and Adamic (2005)) model information "infections" in blogosphere through the analysis of community membership, text similarity, and copying of published URLs between pairs of blogs. How often a link is copied from one blog to another is factored in both determining the influential nodes and in visualization of "infection graphs".

In the framework of various NLP tasks, such as word sense induction and disambiguation, a wealth of computational methods has been developed for the analysis of distributional similarity between words using word co-occurrence patterns. In such methods, contexts of occurrence for each word, comprised by "bags of words" (Schutze, 1998; Widdows and Dorow, 2002) or features based on grammatical dependencies of a given word are compiled into distributional profiles (Hindle, 1990; Pereira et al., 1993; Grefenstette, 1994; Lin, 1998; Pantel and Lin, 2002; Rumshisky and Grinberg, 2009). Similarity between such profiles is computed using vector space (e.g. cosine), set-theoretic (e.g. Dice, Jaccard), graph-based, or information-theoretic similarity measures (e.g. relative entropy, Jensen-Shannon divergence). While such paradigmatic relations between words are captured through distributional similarity, syntagmatic relations between words and their significant collocates are captured through various association scores (e.g. mutual information, t-test scores) (Church and Hanks, 1990; Kilgarriff et al., 2004). Clearly, changing distributional patterns for words related to a particular concept should be taken into account when modeling change in knowledge representation.

Wikipedia has attracted a lot of research both on the analysis of the resulting concept structures (applying both graph-based and word co-occurrence methods) and the use of the resulting resource in NLP tasks. Effectively, Wikipedia provides two resources, a set of hyperlinked articles (the *article graph*) and a taxonomy-like set of categories with the hyponymy- or meronymy-based subsumption relation (the *category graph*, which is allowed to contain cycles and disconnected nodes). A number of studies in recent literature examined the properties of both graphs, and both graphs have been used to measure semantic relatedness of between concepts and to evaluate the overall semantic structure of covered topics. For example, Zlatić et al. (2006)) examined the link structure of the article graph for such properties as degree distributions, growth, topology, reciprocity, clustering, etc. Zesch and Gurevych (2007)) looked at the applicability of standard semantic relatedness measures to the category graph. Bellomi and Bonato (2005)) applied HITS and PageRank algorithms to the article graph to evaluate the importance of each category. Other parameters have also been used to analyze semantic structure of the *article graph*. Holloway et al. (2007)) used the co-occurrence of categories within individual articles to analyze and and visually map semantic interrelations between categories, comparing the resulting graphs for Wikipedia, Britannica and Microsoft Encarta. Buriol et al. (2006)) have looked at the evolution of the article graph over time using timestamps associated with each article.

8. Conclusions and Future Work

The project has demonstrated the potential for a NLP based analysis of long-range historical datasets. Our research has confirmed the validity of our initial basic hypothesis, namely that the relations between textual entities in an encyclopedia can be used as a proxy for relations between corresponding conceptual entities in the minds of educated contemporaries. The resulting software tools for mapping the history of reputations and social networks of the past have a potential to become useful and entertaining educational tools. In the future we plan to extend our research and tools beyond person articles into the realm of more complicated concepts. This will require additional work on the problem of changing taxonomies and disambiguation. Another important extension of our current research will be an analysis of changes in the relations between different domains, rather than between the members within the same domains.

9. Acknowledgments

This research is sponsored in part by a Digital Humanities Start-Up grant from the National Endowment from the Humanities (HD-51128-10). We would like to thank Britannica for providing us with research access to the current, 15th edition. Historical editions were obtained from GoogleBooks scans. We would like thank Jon Orwant from GoogleBooks for his assistance in obtaining the requisite texts.

10. References

Eytan Adar and Lada A. Adamic. 2005. Tracking information epidemics in blogspace. *Web Intelligence, 2005. Proceedings. The 2005 IEEE/WIC/ACM International Conference on*, pages 207–214.

F. Bellomi and R. Bonato. 2005. Network analysis for wikipedia. In *Proceedings of Wikimania*.

K.W. Boyack, R. Klavans, and K. Börner. 2005. Mapping the backbone of science. *Scientometrics*, 64(3):351–374.

J. Brown and P. Reinegen. 1987. Social ties and word-of-mouth referral behavior. *Journal of Consumer Research*, 14(3):350–362.

L.S. Buriol, C. Castillo, D. Donato, S. Leonardi, and S. Millozzi. 2006. Temporal analysis of the wikigraph. In *Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence*, pages 45–51. IEEE Computer Society Washington, DC, USA.

Yun Chi, Shenghuo Zhu, Xiaodan Song, Junichi Tatemura, and Belle L. Tseng. 2007. Structural and temporal analysis of the blogosphere through community factorization. In *KDD '07: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 163–172, New York, NY, USA. ACM.

K. Church and P. Hanks. 1990. Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16:22–29.

J. Coleman, H. Menzel, and E. Katz. 1966. *Medical Innovations: A Diffusion Study*. Bobbs Merrill.

P. Domingos and M. Richardson. 2001. Mining the network value of customers. In *Seventh International Conference on Knowledge Discovery and Data Mining*.

E. Garfield. 1955. Citation indexes for science; a new dimension in documentation through association of ideas. *Science*, 122(3159):108.

E. Garfield. 1972. Citation analysis as a tool in journal evaluation. *Science*, 178(4060):471–479.

P. Gloor, J.S. Krauss, S. Nann, K. Fischbach, D. Schoder, and B. Switzerland. 2008. Web science 2.0: Identifying trends through semantic social network analysis.

G. Grefenstette. 1994. *Explorations in Automatic Thesaurus Discovery*. Kluwer Academic Publishers, Norwell, MA, USA.

D. Gruhl, R. Guha, D. Liben-Nowell, and A. Tomkins. 2004. Information diffusion through blogspace. In *Proceedings of the 13th international conference on World Wide Web*, pages 491–501. ACM New York, NY, USA.

D. Hindle. 1990. Noun classification from predicate-argument structures. In *28th Annual Meeting of the Association for Computational Linguistics*, pages 268–275, Pittsburgh, PA. Association for Computational Linguistics.

T. Holloway, M. Božicevic, and K. Börner. 2007. Analyzing and visualizing the semantic coverage of wikipedia and its authors. *Complexity, Special issue on Understanding Complex Systems*, 12(3):30–40.

A. Kilgarriff, P. Rychly, P. Smrz, and D. Tugwell. 2004. The Sketch Engine. *Proceedings of Euralex, Lorient, France*, pages 105–116.

M. Kimura, K. Saito, and R. Nakano. 2007. Extracting influential nodes for information diffusion on a social network. In *Proceedings of AAAI*, volume 22, page 1371. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999.

J.M. Kleinberg. 1999. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632.

J. Kleinberg. 2003. Bursty and hierarchical structure in streams. *Data Mining and Knowledge Discovery*, 7(4):373–397.

R. Kumar, J. Novak, P. Raghavan, and A. Tomkins. 2005. On the bursty evolution of blogspace. *World Wide Web*, 8(2):159–178.

J. Leskovec, L. Backstrom, and J. Kleinberg. 2009. Memetracking and the dynamics of the news cycle. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 497–506. ACM New York, NY, USA.

D. Lin. 1998. Automatic retrieval and clustering of similar words. *COLING-ACL, Montreal, Canada*.

V. Mahajan, E. Muller, and F. Bass. 1990. New product diffusion models in marketing: A review and directions for research. *Journal of Marketing*, 54(1):1–26.

S. Morris. 2000. Contagion. *Review of Economic Studies*, 67:57–78.

S. Nakajima, J. Tatemura, Y. Hino, Y. Hara, and K. Tanaka. 2005. Discovering important bloggers based on analyzing blog threads. In *Workshop on the Weblogging Ecosystem, 14th International World Wide Web Conference*.

L. Page, S. Brin, R. Motwani, and T. Winograd. 1998. The pagerank citation ranking: Bringing order to the web.

P. Pantel and D. Lin. 2002. Discovering word senses from text. In *Proceedings of ACM SIGKDD02*.

F. Pereira, N. Tishby, and L. Lee. 1993. Distributional clustering of english words. In *Meeting of the Association for Computational Linguistics*, pages 183–190.

D. Rice. 1965. Networks of scientific papers. *Science*, 149:510–515.

E.M. Rogers. 1979. Network analysis of the diffusion of innovations. *Perspectives on social network research*, pages 137–164.

A. Rumshisky and V. A. Grinberg. 2009. Using semantics of the arguments for predicate sense induction. In *Proceedings of 5th International Conference on Generative Approaches to the Lexicon (Gl2009)*, Pisa, Italy.

H. Schutze. 1998. Automatic word sense discrimination. *Computational Linguistics*, 24(1):97–123.

D. Widdows and B. Dorow. 2002. A graph model for unsupervised lexical acquisition. In *19th International Conference on Computational Linguistics*, pages 1093–1099, Taipei, Taiwan.

H. P. Young. 1998. *Individual Strategy and Social Structure: An Evolutionary Theory of Institutions*. Princeton.

T. Zesch and I. Gurevych. 2007. Analysis of the Wikipedia category graph for NLP applications. In *Proc of NAACL-HLT 2007 Workshop: TextGraphs*, volume 2.

V. Zlatić, M. Božičević, H. Štefančić, and M. Domazet. 2006. Wikipedias: Collaborative web-based encyclopedias as complex networks. *Physical Review E*, 74(1).

Building a Corpus of Old Czech

Jirka Hana,[1] **Boris Lehečka,**[2] **Anna Feldman,**[3] **Alena Černá,**[2] **Karel Oliva**[2]

[1]Charles University, MFF, Prague, Czech Republic
[2]The Academy of Sciences of the Czech Republic, Institute of the Czech Language, Prague, Czech Republic
[3]Montclair State University, Montclair, NJ, USA

Abstract

In this paper we describe our efforts to build a corpus of Old Czech. We report on tools, resources and methodologies used during the corpus development as well as discuss the corpus sources and structure, the tagset used, the approach to lemmatization, morphological analysis and tagging. Due to practical restrictions we adapt resources and tools developed for Modern Czech. However, some of the described challenges, such as the non-standardized spelling in early Czech and the form and lemma variability due to language change during the covered time-span, are unique and never arise when building synchronic corpora of Modern Czech.

Keywords: Old Czech; Corpus; Morphology

1. Introduction

This paper describes a corpus of Old Czech and the tools, resources and methodologies used during its development. The practical restrictions (no native speakers, limited amount of available texts and lexicons, limited funding) preclude the traditional resource-intensive approach used in the creation of corpora for large modern languages. However, many high-quality tools, resources and guidelines exist for Modern Czech, which is in many aspects similar to Old Czech despite 500 years of development. This means that most tools, etc. do not need to be developed from scratch, but instead can be based on tools for Modern Czech.

Our paper is structured as follows. We outline the relevant aspects of the Czech language and compare its Modern and Old forms (§2.). We describe the sources and basic attributes of the corpus (§3.); lemmas and tagset used in annotation (§4.); semi-manual lemmatization (§5.); and finally, resource light morphological analysis and tagging based on Modern Czech and its more resource-intensive improvement (§6.).

2. Czech

Czech is a West Slavic language with significant influences from German, Latin and (in modern times) English. It is a fusional (inflective) language with rich morphology, a high degree of homonymy of endings and so-called free word-order.

2.1. Old Czech

As a separate language, Czech forms at the end of the 10th century AD. However, the oldest surviving written documents date to the early 1200's. The term Old Czech (OC) usually refers to the language as used roughly between 1150 and 1500. It is followed by Humanistic Czech (1500-1650), Baroque Czech (1650-1780) and then Czech of the so-called National Revival. Old Czech was significantly influenced by Old Church Slavonic, Latin and German.

2.2. Modern Czech

Modern Czech (MC) is spoken by roughly 10 million speakers, mostly in the Czech Republic. For a more detailed discussion, see for example (Naughton, 2005; Short, 1993; Janda and Townsend, 2002; Karlík et al., 1996). For historical reasons, there are two variants of Czech: Official (Literary, Standard) Czech and Common (Colloquial) Czech. The official variant is based on the 19th-century resurrection of the 16th-century Czech. The two variants are influencing each other, resulting in a significant amount of irregularity, especially in morphology. The Czech writing system is mostly phonological.

2.3. Differences

Old Czech differs from Modern Czech in many aspects, including orthography, phonology, morphology and syntax. Some of the changes occurred during the period of Old Czech. Providing a systematic description of differences between Old and Modern Czech is beyond the scope of this paper. Therefore, we just briefly mention a few illustrative examples. For a more detailed description see (Vážný, 1964; Dostál, 1967; Mann, 1977).

2.3.1. Phonology and Spelling

Examples of some of the more regular sound changes between OC and MC can be found in Table 1. Moreover, the difference in the pronunciation of *y* and *i* is lost, with *y* being pronounced as *i* (however, the spelling still in most cases preserves the original distinction). In addition to these linguistic changes, the orthography develops as well; for more details, see (Křístek, 1978; Kučera, 1998).

2.3.2. Nominal Morphology

The nouns of OC have three genders: feminine, masculine, and neuter. In declension they distinguish three numbers: singular, plural, and dual, and seven cases: nominative, genitive, dative, accusative, vocative, locative and instrumental. Vocative is distinct only for some nouns and only in singular.

During the Old Czech period, the declension system moves from a noun-to-paradigm assignment based on the stems to an assignment based on gender. The dual number is replaced by plural, e.g., OC: *s jedinýma dvěma děvečkama* vs. MC: *s jedinými dvěma děvečkami* 'with the only two maids'. In MC, the dual number survives only in declension of a few words, such as the paired names of parts of

change during OC	later change	example		
ú > ou		*múka*	*> mouka*	'flour'
'ú > í		*kľúč*	*> klíč*	'key'
sě > se		*sěno*	*> seno*	'hay'
ó > uo	*> ů*	*kóň*	*> kuoň > kůň*	'horse'
'ó > ie	*> í*	*koňóm*	*> koniem > koním*	'horse$_{dat.pl}$'
šč > šť		*ščúr*	*> štír*	'scorpion'
čs > c		*čso*	*> co*	'what'

Table 1: Examples of sound/spelling changes from OC to MC

category		Old Czech	Modern Czech
infinitive		péc-i	péc-t 'bake'
present	1sg	pek-u	peč-u
	1du	peč-evě	–
	1pl	peč-em(e/y)	peč-eme
	:		
imperfect	1sg	peč-iech	–
	1du	peč-iechově	–
	1pl	peč-iechom(e/y)	–
	:		
sigm. aorist	1sg	peč-ech	–
	1du	peč-echově	–
	3du	peč-esta	–
	1pl	peč-echom(e/y)	–
	:		
imperative	2sg	pec-i	peč
	2du	pec-ta	–
	2pl	pec-te	peč-te
	:		
verbal noun		peč-enie	peč-ení

Table 2: A fragment of the conjugation of the verb *péci/péct* 'bake' (OC based on (Dostál, 1967, 74-77))

the body and the agreeing attributes. In Common Czech the dual plural distinction is completely neutralized. On the other hand, MC distinguishes animacy in masculine gender, while this distinction starts to emerge only in late OC.

2.3.3. Verbal Morphology

The system of verbal forms and constructions was far more elaborate in OC than in MC. Many forms disappeared, e.g., aorist and imperfect (simple past tenses), supine; and some became archaic, e.g., verbal adverbs, plusquamperfectum). All dual forms are no longer in MC (OC: *Herodes s Pilátem sě smířista*; MC: *Herodes s Pilátem se smířili* 'Herod and Pilate reconciled'). See Table 2 for an example. The periphrastic future tense is stabilized; both *bude slúžil* and *bude slúžiti* used to mean 'will serve', but only the latter form is possible now.

3. Old Czech Corpus

The manuscripts and incunabula written in Old Czech are being made accessible by the Institute of Czech Language. They are transcribed and included into the Old-Czech Text Bank, which is a part of the Web Vocabulary.[1]

So far, 124 Old Czech documents, or 2.8M tokens, have been processed and incorporated into the Old-Czech Text Bank.[2] Most of them date to 1400's, the period from which most documents survived. The corpus is not balanced in respect to the periods and genres of the included documents. Nevertheless, currently, it contains a variety of documents, including liturgical, legal and medical texts, travel books, sermons, prayers, deeds, chronicles, songs, etc. Our goal is to eventually incorporate all surviving documents, including their variants. There are at least 1239 documents, as this is the number of sources of the (StčS, 1968) Old Czech dictionary.

The Old Czech spelling varied significantly. First, the period covers about 350 years, so spelling changes are expected. Second, spelling at this time was not standardized; therefore, the same word can have many different spelling variants even at the same time. Obviously, this causes many practical problems when working with the Old Czech data. For this reason, we transcribe all documents using the spelling conventions of Modern Czech, while preserving the specific features of Old Czech. This standardizes the graphemic representation of words with variant spelling, e.g., *czieſta*, *czėſta*, *cyeſta* are all represented as *cěsta*, MC: *cesta* 'path'. It also makes the texts accessible to users without philological background. For more details, see (Lehečka and Voleková, 2010).

4. Lemmas and tagset

4.1. Principles of lemmatization

Similarly to many modern language corpora, our goal is to provide information about lemma for each word. By lemma (canonical or citation form) we mean a form distinguished from a set of all forms related by inflection. Lemmas are chosen by convention (e.g., nominative singular for nouns, infinitive for verbs). As lemmas abstract away from the inflection of words, they can be useful, for example, in searching the corpus, especially for lexicography.

However, as the language changed during the period covered in the corpus, so did lemmas. This means that the same word might be assigned different lemmas in different texts (for example, *kóň*, *kuoň*, *kůň* are different historical variants of the same lemma). In some cases, a user might be interested in a particular historical variant of a lexeme, but in other they might want to search for all historical variants. As a solution, we use two levels of lemmas: (1) a traditional lemma phonologically consistent with the particular

[1] See http://vokabular.ujc.cas.cz/banka.aspx.

[2] http://vokabular.ujc.cas.cz/texty.aspx?id=STB

form(s) in the text; (2) a hyperlemma, reflecting phonology around 1300. Thus, for example, the hyperlemma *kóň* would correspond to lemmas *kóň*, *kuoň*, *kůň*.

In addition, we allow a single form token to be assigned multiple lemmas and hyperlemmas and possibly, morphological tags even in a disambiguated annotation. This is used for cases when even context does not help to select a single value.

The corpus manager and viewer,[3] has been modified to support these specific features of the historical corpus.

4.2. Tagset

We adopted the tag system originally developed for Modern Czech (Hajič, 2004). Every tag is represented as a string of 15 symbols each corresponding to one morphological category (2 positions out of 15 are not used). Features not applicable for a particular word have a N/A value. For example, when a word is annotated as `AAFS4---2A---` it is an adjective (`A`), long form (`A`), feminine (`F`), singular (`S`), accusative (`4`), comparative (`2`), not-negated (`A`). The tagset has more than 4200 tags; however, only about half of them occur in a 500M token corpus.

The modification for Old Czech is quite straightforward. No additional tag positions are added, but the last slot distinguishing stylistic variants is not used. We add values for categories not present in MC (e.g., aorist, imperfect).

In addition to changes motivated by language change, we avoid using wildcard values (symbols representing a set of atomic values, e.g., H for feminine or neuter gender) for reason outlined in (Hana and Feldman, 2010). While wildcards might lead to better tagging performance, they provide less information about the word, which might be needed for linguistic analysis or an NLP application. In addition, it is trivial to translate atomic values to wildcards if needed. The Old-Czech tagset contains only wildcards covering all atomic values (denoted by X for all applicable positions). There are no wildcards covering a subset of atomic values. Forms that would be tagged with a tag containing a partial wildcard in Modern Czech are regarded as ambiguous.

5. Semi-manual lematization

We perform partial manual lemmatization of the corpus, exploiting Zipf's law (Zipf, 1935; Zipf, 1949): the 2,000 most frequent form types cover 75% of 2.8M tokens of the corpus. We manually assign lemmas to these forms, taking into account homonymy and lemma variants. The words in the corpus are then assigned candidate lemmas based on this list.

In the future, we are planning to increase the recall of this method by considering prefixes. For example *spomoci*, *přemoci*, *dopomoci přěmoci* all have a low frequency and are thus not covered by the manually lemmatized list of frequent forms. However, they all are derived by prefixation from the word *moci* 'can', which is much more frequent and is thus covered. Also, we would like to consider regular sound change. For example, applying sound change *’ě >e*, one could translate the lemma *cesta* 'path' of *cestu* to the lemma *cěsta* of the less frequent *cěstu.*

6. Resource light morphology

The practical restrictions (no native speakers, limited corpora and lexicons, limited funding) make Old Czech an ideal candidate for the resource-light crosslingual method that we have been developing (Feldman and Hana, 2010). The first results were reported in (Hana et al., 2011). In this section, we describe the basics of our approach and some of its extensions.

The main assumption of our method (Feldman and Hana, 2010) is that a model for the target language can be approximated by language models from one or more related source languages and that the inclusion of a limited amount of high-impact and/or low-cost manual resources is greatly beneficial. We are aware of the fact that all layers of the language have changed during the last 500+ years, including phonology and spelling, syntax and vocabulary. Even words that are still used in MC often appear with different distributions, with different declensions, with different gender, etc.

6.1. Materials

Our MC *training* corpus is a portion (700K tokens) of the Prague Dependency Treebank (PDT, Hajič et al. (2006)). The corpus contains texts from daily newspapers, business and popular scientific magazines. It is manually morphologically annotated.

Several steps (e.g., lexicon acquisition) of our method require a plain text corpus. We used texts from the Old-Czech Text Bank. The corpus is significantly smaller than the corpora we used in other experiments (e.g., 39M tokens for Czech or 63M tokens for Catalan (Feldman and Hana, 2010)).

A small portion (about 1000 words) of the corpus was manually annotated for testing purposes.

6.2. Tools

6.2.1. Tagger

We use TnT (Brants, 2000), a second order Markov Model tagger. The language model of such a tagger consists of emission probabilities (corresponding to a lexicon with usage frequency information) and transition probabilities (roughly corresponding to syntax rules with strong emphasis on local word-order). We approximate the emission and transition probabilities by those trained on a modified corpus of a related language.

6.2.2. Resource-light Morphological Analysis

The *Even* tagger described in the following section relies on a morphological analyzer. While it can use any analyzer, to stay within a resource light paradigm, we use our resource-light analyzer (Hana, 2008; Feldman and Hana, 2010), which relies on a small amount of manually or semi-automatically encoded morphological details. In addition to modules we used for other languages, we also include an analyzer for Modern Czech which is used as a safety-net in parallel to an ending-based guesser.

[3] See `http://sourceforge.net/projects/corpman/` for the current version

The results of the analyzer are summarized in Table 3. They show a similar pattern to the results we have obtained for other fusional languages. As can be seen, morphological analysis without any filters (the first two columns) gives good recall but also very high average ambiguity. When the automatically acquired lexicon and the longest-ending filter (analyses involving the longest endings are preferred) are used, the ambiguity is reduced significantly but recall drops as well. As with other languages, even for OC, it turns out that the drop in recall is worth the ambiguity reduction when the results are used by our MA-based taggers.

Lexicon & leo	no		yes	
	Recall	Ambiguity	Recall	Ambiguity
Overall	96.9	14.8	91.5	5.7
Nouns	99.9	26.1	83.9	10.1
Adjectives	96.8	26.5	96.8	8.8
Verbs	97.8	22.1	95.6	6.2

Table 3: Evaluation of the morphological analyzer on Old Czech

6.3. Experiments

We describe three different taggers:

1. a TnT tagger using modified MC corpus as a source of both transition and emission probabilities (section 6.3.1.);
2. a TnT tagger using modern transitions but approximating emissions by a uniformly distributed output of a morphological analyzer (MA) (sections 6.2.2. and 6.4.); and
3. a combination of both (section 6.5.).

6.3.1. Translation Model

Modernizing OC and Aging MC We modify the MC corpus so that it looks more like the OC just in the aspects relevant for morphological tagging. These modifications include translating the tagset, reversing phonological/graphemic changes, etc. Unfortunately, even this is not always possible or practical. For example, historical linguists usually describe phonological changes from old to new, not from new to old.[4] In addition, it is not possible to deterministically translate the modern tagset to the older one. So, we modify the MC training corpus to look more like the OC corpus (the process we call 'aging') and also the target OC corpus to look more like the MC corpus ('modernizing').

Creating the Translation Tagger Below we describe the process of creating a tagger. As an example we discuss the details for the *Translation* tagger. Figure 1 summarizes the discussion.

1. Aging the MC training (annotated) corpus:
   - MC to OC tag translation:
     Dropping animacy distinction (OC did not distinguish animacy).
   - Simple MC to OC form transformations:
     E.g., modern infinitives end in *-t*, OC infinitives ended in *-ti*;
     (we implemented 3 transformations)
2. Training an MC tagger. The tagger is trained on the result of the previous step.
3. Modernizing an OC plain corpus. In this step we modernize OC forms by applying sound/graphemic changes such as those in Table 1. Obviously, these transformations are not without problems. First, the OC-to-MC translations do not always result in correct MC forms; even worse, they do not always provide forms that ever existed. Sometimes these transformations lead to forms that do exist in MC, but are unrelated to the source form. Nevertheless, we think that these cases are true exceptions from the rule and that in the majority of cases, these OC translated forms will result in existing MC words and have a similar distribution.
4. Tagging. The modernized corpus is tagged with the aged tagger.
5. Reverting modernizations. Modernized words are replaced with their original forms. This gives us a tagged OC corpus, which can be used for training.
6. Training an OC tagger. The tagger is trained on the result of the previous step. The result of this training is an OC tagger.

		Transl	Even	TranslEven
All	Full:	70.6	67.7	74.1
	SubPOS	88.9	87.0	90.6
Nouns	Full	63.1	44.3	57.0
	SubPOS	99.3	88.6	91.3
Adjs	Full:	60.3	50.8	60.3
	SubPos	93.7	87.3	93.7
Verbs	Full	47.8	74.4	80.0
	SubPOS	62.2	78.9	86.7

Table 4: Performance of various tagging models on major POS categories (in % on full tags and the SubPOS position).

The results of the translation model are provided in Table 4 (across various POS categories). The Translation tagger is already quite good at predicting the POS, SubPOS (Detailed POS) and number categories. The most challenging POS category is the category of verbs and the most difficult feature is case. Based on our previous experience with other fusional languages, getting the case feature right is always challenging. Even though case participates in syntactic agreement in both OC and MC, this category is more idiosyncratic than, say, person or tense. Therefore, the MC

[4] Note that one cannot simply reverse the rules, as in general, the function is not a bijection.

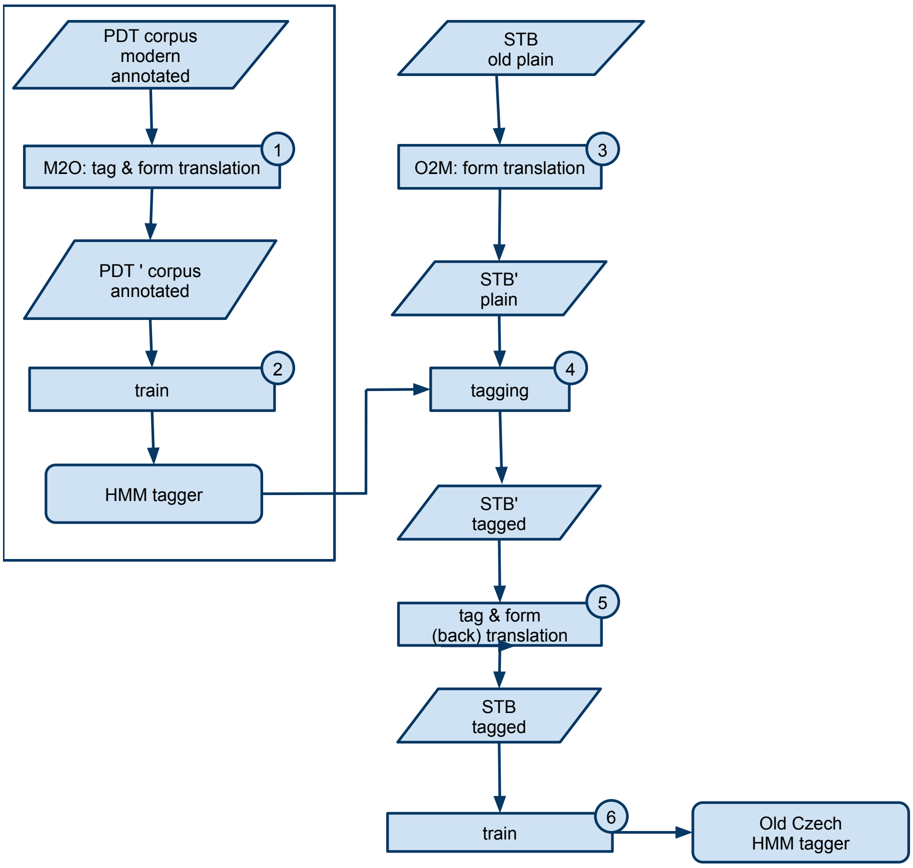


Figure 1: Schema of the Translation Tagger

syntactic and lexical information provided by the translation model might not be sufficient to compute case correctly. One of the solutions that we explore in this paper is approximating the OC lexical distribution by the resource-light morphological analyzer (see section 6.4.).

While most nominal forms and their morphological categories (apart from dual) survived in MC, OC and MC departed in verbs significantly. Thus, for example, three OC tenses disappeared in MC and other tenses replaced them. These include the OC two aorists, supinum and imperfectum. The transgressive forms are almost not used in MC anymore either. Instead MC has periphrastic past, periphrastic conditional and also future. In addition, these OC verbal forms that disappeared in MC are unique and non-ambiguous, which makes it even more difficult to guess if the model is trained on the MC data. The tagger, in fact, has no way of providing the right answer. In the subsequent sections we use the morphological analyzer described above to address this problem. Recall that our morphological analyzer uses only very basic hand-encoded facts about the target language.

6.4. Even Tagger

The *Even* tagger (see Figure 2) approximates emissions by uniformly (evenly) distributing the tags output by our morphological analyzer. The transition probabilities are based on the Aged Modern Czech corpus (result of step 2 of Figure 1). This means that the transitions are produced during the training phase and are independent of the tagged text. However, the emissions are produced by the morphological analyzer on the basis of the tagged text during tagging.

The overall performance of the Even tagger drops down, but it improves on verbs significantly. Intuitively, this seems natural, because there is relatively small homonymy among many OC verbal endings (see Table 2 for an example) so they are predicted by the morphological analyzer with low or even no ambiguity.

6.5. Combining the Translation and Even Taggers

The *TranslEven* tagger is a combination of the Translation and Even models. The Even model clearly performs better on the verbs, while the Translation model predicts other categories much better. So, we decided to combine the two models in the following way. The Even model predicts verbs, while the Translation model predicts the other categories. The TranslEven Tagger gives us a better overall performance and improves the prediction on each individual position of the tag. Unfortunately, it slightly reduces the performance on nouns (see Table 4).

6.6. Discussion

OC and MC departed significantly over the 500+ years, at all language layers, including phonology, syntax and vocabulary. Words that are still used in MC are often used with different distributions and have different morphological forms from OC.

An additional difficulty of this task arises from the fact that our MC and OC corpora belong to different genres. While the OC corpus includes among others poetry, chronicles,

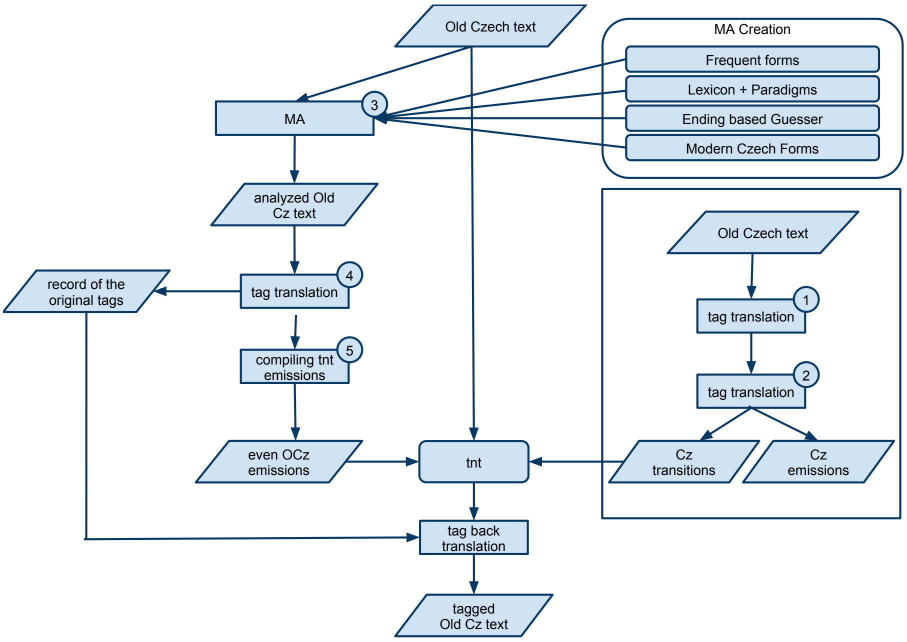


Figure 2: Schema of the MA Based Even Tagger

medical and liturgical texts, the MC corpus is mainly comprised of newspaper texts. We cannot possibly expect a significant overlap in lexicon or syntactic constructions. For example, the cookbooks contain a lot of imperatives and second person pronouns which are rare or non-existent in the newspaper texts.

Even though our tagger does not perform as the state-of-the-art tagger for Czech, the results are already useful. Remember that the tag is a combination of 12 morphological features[5] positions out and if only one of them is incorrect, the whole positional tag is marked as incorrect. So, the performance of the tagger (74%) on the whole tag is not as low in reality. For example, if one is only interested in detailed POS (i.e., the SubPOS position, about 70 values) information the performance of our system is over 90%.

6.7. Improving morphology

To stay within the resource-light paradigm, the tagger and the morphological analyzer described in the previous section intentionally avoid resources that are unlikely to be available for a wide range of languages. However, for practical reasons it makes sense to develop tools that make use of any resources available for Old Czech.

In this section we describe a morphological analyzer improving upon the analyzer from §6.2.2. by incorporating a list of known sound changes, such as those in Table 1. We have used these changes to "translate" Old Czech words into modern Czech. Such words were analyzed by the Modern Analyzer and the result was then translated back to Old Czech. As most of the sound changes have many exceptions, we used all subsets of those rules (including an empty set), possibly assigning more than one Modern Czech equivalent to an Old Czech word. This means the translation overgenerates, potentially assigning a number of forms exponential to the number of rules. Nevertheless, in practice this is not a problem. For example, the Old Czech word *kacířův* 'heretic's' is assigned 18 different modernized equivalents (see Figure 3). However, the modern Czech analyzer recognizes only *kacířův*, the correct translation. Therefore, *kacieřóv* will be correctly analyzed as possessive adjective. Most forms have fewer than 18 modernized equivalents. The results of the analyzer incorporating a module used in such translations are given in Table 5. One can see that the results improve on nearly all categories and POS. A tagger using this analyzer achieves a similar improvement.

kacířův
kacířév
kacieřiev
kacieřóv
kacéřóv
kacieřuov

kacéřiev
kacéřív
kacířív
kacéřův
kacieřív
kacířóv

kacířiev
kacéřév
kacéřuov
kacieřév
kacieřův
kacířuov

Figure 3: Modernized equivalents of an Old Czech word *kacieřóv*, MC: *kacířův* 'heretic's'

Lexicon & leo	no		yes	
	Recall	Ambiguity	Recall	Ambiguity
Overall	97.1	7.3	94.2	4.2
Nouns	99.0	10.0	90.6	5.8
Adjectives	96.8	17.0	96.8	8.7
Verbs	97.8	10.9	97.8	3.3

Table 5: Evaluation of the morphological analyzer using sound change rules

[5] The tag has 15 positions, but two of them are not used and we do not evaluate on the variant position as its values are to a great extent arbitrary.

6.8. Conclusion

We have presented a corpus of Old Czech currently under development. Many of the tools used during the development process are resource-light and/or rely on resources developed for Modern Czech. While the results (for example of the taggers) are significantly lower than the corresponding results for Modern Czech, they are achieved with a fraction of resources and for many practical applications are already good enough.

Acknowledgments

This research was supported by the Grant Agency of the Czech Republic (projects ID: P406/10/1140 and P406/10/P328) and by the U.S. NSF grants #0916280, #1033275. Also, this work has been using language resources developed and/or stored and/or distributed by the LINDAT-Clarin project of the Ministry of Education of the Czech Republic (project LM2010013). We would also like to thank the two anonymous reviewers for their comments.

7. References

Thorsten Brants. 2000. TnT – A Statistical Part-of-Speech Tagger. In *Proceedings of ANLP-NAACL*, pages 224–231.

Antonín Dostál. 1967. *Historická mluvnice česká II – Tvarosloví. 2. Časování [Historical Czech Grammar II – Morphology. 2. Conjugation]*. Praha.

Anna Feldman and Jirka Hana. 2010. *A resource-light approach to morpho-syntactic tagging*. Rodopi, Amsterdam/New York, NY.

Jan Hajič, Jarmila Panevová, Eva Hajičová, Petr Sgall, Petr Pajas, Jan Štěpánek, Jiří Havelka, Marie Mikulová, Zdeněk Žabokrtský, and Magda Ševčíková-Razímová. 2006. *Prague Dependency Treebank 2.0*. Linguistic Data Consortium, Philadelphia, PA, USA.

Jan Hajič. 2004. *Disambiguation of Rich Inflection: Computational Morphology of Czech*. Karolinum, Charles University Press, Praha.

Jirka Hana and Anna Feldman. 2010. A positional tagset for Russian. In *Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC 2010)*, pages 1278–1284, Valletta, Malta. European Language Resources Association.

Jirka Hana, Anna Feldman, and Katsiaryna Aharodnik. 2011. A low-budget tagger for old czech. In *Proceedings of the 5th ACL-HLT Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*, pages 10–18, Portland, OR, USA, June. Association for Computational Linguistics.

Jirka Hana. 2008. Knowledge- and labor-light morphological analysis. *OSUWPL*, 58:52–84.

Laura A. Janda and Charles E. Townsend. 2002. Czech. `http://www.seelrc.org:8080/grammar/mainframe.jsp?nLanguageID=2`.

Petr Karlík, Marek Nekula, and Zdenka Rusínová. 1996. *Příruční mluvnice češtiny [Concise Grammar of Czech]*. Nakladatelství Lidové noviny, Praha.

Václav Křístek. 1978. *Malý staročeský slovník [Short Old Czech Dictionary]. Část Staročeské pravopisné systémy [Part Old Czech Spelling Systems]*. Praha.

Karel Kučera. 1998. Vývoj účinnosti a složitosti českého pravopisu od konce 13. století do konce 20. století. *Slovo a slovesnost*, 59:178–199.

Boris Lehečka and Kateřina Voleková. 2010. (Polo)automatická počítačová transkripce [(Semi)automatic computational transcription]. In *Proceedings of the Conference Dějiny českého pravopisu (do r. 1902) [History of the Czech spelling (before 1902)]*.

Stuart E. Mann. 1977. *Czech Historical Grammar*. Hamburg: Buske.

James Naughton. 2005. *Czech: An Essential Grammar*. Routledge, Oxon, Great Britain and New York, NY, USA.

David Short. 1993. Czech. In Bernard Comrie and Grevilled G. Corbett, editors, *The Slavonic Languages*, Routledge Language Family Descriptions, pages 455–532. Routledge.

StčS. 1968. *Staročeský slovník [Old Czech dictionary]. Část Úvodní stati, soupis pramenů a zkratek. [Part Introduction, list of sources and abbreviations]*. Praha.

Václav Vážný. 1964. *Historická mluvnice česká II – Tvarosloví. 1. Skloňování [Historical Czech Grammar II – Morphology. 1. Declension]*. Praha.

George K. Zipf. 1935. *The Psychobiology of Language*. Houghton-Mifflin.

George K. Zipf. 1949. *Human Behavior and the Principle of Least-Effort*. Addison-Wesley.

Building an electronic dictionary of Old Polish on the base of the paper resource

Agnieszka Mykowiecka, Piotr Rychlik and Jakub Waszczuk

Institute of Computer Science, Polish Academy of Sciences
ul. Jana Kazimierza 5, 01-248 Warsaw, Poland
agn@ipipan.waw.pl,rychlik@ipipan.waw.pl, waszczuk.kuba@gmail.com

Abstract

In this paper we present a process of converting an existing dictionary of Old Polish into LMF (Lexical Markup Framework) format. We discuss problems related to the transformation of a resource build to be used as a paper book into its electronic version. We describe the subsequent stages of the process consisting in scanning the paper source followed by OCR and correction of its results, converting the data into LMF format and enriching the dictionary with information which was given indirectly or which can be obtained from other sources. In particular we describe the method of assigning part of speech names to lexicon entries.

1. Introduction

One of the effects of the rapid development of new technologies is the substantial change of the way in which not only contemporary but also historic documents can be accessed. Archival documents which are too valuable to be accessed frequently, now, in digital form can be read (either on remote or local computers) by many more readers without influencing their original paper version. Although the process of mass digitization is still at the early stage (a very small part of documents is converted into digital form) there is already a lot of documents available on the servers of various kinds of digital libraries. With the increasing availability, the interest in reading old texts also increases, so there rose a need for developing tools and resources to process them. Old text are interesting not only for experts, i.e. people who have knowledge to read and understand them in their original form, but also for general users (hobbyists, students). For them, the resources of the first need are electronic dictionaries capable of presenting users contemporary equivalents of the words which are no longer in use or which now have different meanings. For many languages, like for Polish, this type of language resources is sill lacking. In spite of the rapid development of natural language processing tools, historical linguists working on Polish still work mostly on paper documents, or at best on the digital scans which cannot be easily searched. The problem of access concerns also dictionaries of Old Polish. *The Dictionary of the XVII and the first part of the XVIII century* (PAN, 1996) is publicly available as a web page but is far from being finished. *The Dictionary of the XVI century* (Bąk et al., 1966 2002) has been recently made partially accessible in a djVu searchable form, but still the ease of its usage is far from being satisfactory. Our goal was to prepare a resource which can be used by a general audience interested in reading all texts. The desirable features of the lexicon are a coverage of a long time period, ease of use (i.e. clarity of the interface) and availability of different search methods. The dictionary was meant to be easily extended because in time we plan to enlarge it with the content of other existing resources if their authors agree to make their data widely available.

The problem of elaborating a dictionary of a particular language variant can be addressed in many ways. When considering the source of data, the prevailing tendency is now building lexicons on the basis of specially collected text corpora. But, at the moment there are no corpora with Old Polish texts stored in the form and with the quality which would be appropriate for the purpose, and what is even more important, there are no resources needed to construct lexicon entries' definitions. Thus, we decided to check how hard and effective would be a process of converting an existing old paper dictionary into a fully operational electronic form. In particular, we wanted to test whether at some stages of conversion we can use existing language tools elaborated for Modern Polish as such a dictionary contains not only historical but also Modern Polish material. As a starting point we choose a dictionary which was prepared 43 years ago by Stefan Reczek — „Quick Reference Dictionary of the former Polish language" (Reczek, 1968). This resource is of a moderate size (about 700 pages) and contains words which differ form their current equivalents together with information on their meaning and examples of their usage. They were selected subjectively by its author on the basis of their popularity in Polish from XIII to XVIII centuries. Our goal was to gradually convert the original text into an electronic dictionary to be used mainly by non-specialists.

The research was conducted within the area of a large national project aimed at creating a universal, open hosting and communication platform for network knowledge resources for science, education and open information society (www.synat.pl).

The process of dictionary creation consisted of four main steps:

- scanning the paper source followed by OCR and correction of its results;
- defining the content of the dictionary entry and its description in therms of the LMF format (Lexical Markup Framework, http://www.lexicalmarkup framework.org, (Francopoulo et al., 2009))
- converting the data into LMF format,
- enriching the dictionary with information which was

given in the paper version indirectly or which can be obtained from other sources.

2. OCR Stage

The first step of the process was to scan and OCR the text. The font used for dictionary typesetting was a contemporary one, so the scanned text could be decoded using a standard OCR engine. As the program was not in any way adopted for old vocabulary, there were more than usual non existing words recognized (a lot of errors concerned Polish diacritics partly because they changed and partly because they are always hard to recognize and distinguish from accidental paper defects or dirt). Some errors within lexicon entries could be corrected basing on the information that they are lexically ordered. There were also some systematic errors in recognizing bold face (used to distinguish lexicon entries) and « » characters, which were used to mark beginning and ending of a definition section. After an automatic correction of the most common errors, all entries have been manually verified.

Below there is an example in which one lexicon entry is cited. The meaning of the entry is included within the double angle quotation marks « ». Next, after a colon, an example from the original old text is given followed by the source identifier. In this case, after a nominal form of a noun also an ending of a genitive form is given.

(1) **paciep, -i** *«ciemność, mroczność»: Skoro wynidzie jego planeta z paciepi WPot.*

[**paciep, paciep**$_{gen}$ *«darkness»: When his planet goes out of the darkness. Wacław Potocki.*]

3. Lexicon Entry

The content of the lexicon entry was defined as follows:

- entry heading,
- homonym number,
- orthographic variants,
- grammatical information (POS, subcategorization information),
- word forms and their morphological description,
- word sense number,
- domain qualifier,
- style qualifier,
- definition or contemporary equivalent,
- usage examples,
- multi-word expressions, idioms
- source identification.

The problem with using this schema was twofold. First, not all information which was decided to be important is present within the original dictionary. Second, in the paper version some information is presented in a way which is difficult or even not possible to convert automatically.

As it can already be seen in (1), the typical Reczek's lexicon entry does not contain grammatical information. Information on domain or style qualifier is also given only for very few entires. Like most paper dictionaries it does not contain inflected forms, which in Polish can differ significantly from lemmas. However, this information is available for selected items and should be maintained in the conversion process. Missing information will have to be added later.

The additional problem was to decide on the exact ways of representing all selected information using the LMF format. As Old Polish was much less standardized than Modern, the dictionary contains a lot of variants which were used simultaneously or in different time periods. In this aspect this resource is then similar to dialect dictionaries (like that presented in (de Vriend et al., 2006)). In connection with the fact that Polish is highly inflected language, this made the process of adapting the LMF standard not so straightforward (although some attention to describing inflectional features was also already paid, e.g. (Romary et al., 2004)).

The definition of a dictionary structure together with some preliminary decisions concerning translation rules are described in (Mykowiecka et al., 2011). Here we focus on the conversion process itself, i.e. translating information given in a human readable form into a structured one and the first stage of adding information which is given indirectly or which can be inferred by analyzing other resources, i.e. assigning part of speech names.

4. Interpretation rules

In Reczek's dictionary most lexical entries have a simple internal structure of the form: the name of the entry given in bold is followed by a list of numbered definitions and citations. (In case a list has only one element it is not numbered.) One example of a dictionary definitions was already given in (1).

The conversion of the main entry elements into LMF format is straightforward, but about 20% of entries contain various kinds of additional information coded in a way which is very hard to be processed by a machine.

Some entries include different orthographic variants of old words, see (2). All these variants are given as separate <FormRepresentation> elements inside the <Lemma> tag.

(2) **almaryja, olmaryja** «skrzynia, skrzynka, szkatułka»: *Siedzą sobie w kącie jako olmaryje SPetr.*

[**almaryja, olmaryja** «chest, box, casket»: *They sit in the corner as the boxes. Sebastian Petrycy.*]

There are also some entries which consist of more than one word, see (3). In these cases a list of components is defined indicating lexicon entries describing individual words. Unfortunately, for many such elements the relevant lexicon entries cannot be found. There are two main reasons of this situation. The word can happen to be identical to a contemporary one and was omitted by the author because it does not need explanation. The second type of situation concerns inflected forms of words which are in the dictionary but cannot be easily identified. In the example (3), for the first word *chcenie* there is no lexical entry in the Reczek's dictionary, as the word itself exists still in colloquial Polish as an equivalent of the word *chęć* 'will'. The second word also is still in use, but its meaning changed a bit (form ‘a drink’ into ‘drinking’) and it is described in the dictionary under the basic form – *picie*. To make an electronic resource more complete, we add new entries for all these elements (in this case for *chcenie* and *picia*). In the next stage of a dictionary development we will try to link them to their possible base forms using a version of the Levenstein distance algorithm taking into account specificity of Polish inflection rules. If a multiword expression is given as one of the orthographic variants, it is translated into a separate entry, as there can exists only one <ListOfComponents> for an entry.

(3) **chcenie picia** «pragnienie»: *W chceniu picia mojego napawali mie octem PFl.*

[**chcenie picia** «thirst»: *In my thirst they gave me vinegar to drink. Psałterz floriański.*]

The specific group of multiword entries are reflexive verbs. Reflexive verbs have a reflexive marker "się", e.g. *nudzić się* ‘to be bored’. In these cases the <ListOfComponents> element is not created. Reflexive and non-reflexive forms of a verb are placed in two separate dictionary entries. If these two forms have the same meaning, the reference to the non-reflexive form is given in the <RelatedForm> element inside the LMF structure representing the reflexive verb entry.

Yet another case when a multiword expression can be a candidate for a lexicon entry is illustrated in (4). In this example in place of a phrase, the subcategorization information for a verb is given. Such multiword expressions are recognized by comparing the text to the standard “case identifying” questions and in these cases the information is stored inside the <SyntacticBehaviour> element not as a list of components.

(4) **miarkować co z czym** «porównywać, zestawiać»: Swoje z nieprzyjacielską potęgę miarkując, najemnych ludzi zaciągnął BKrom;

[**miarkować co z czym** «to compare sth with sth»: After comparing his power with the enemy, he hired mercenaries. Marcin Kromer Chronicles translated by Marcin Błażewski.]

Form variants can also be given in a different way than it was shown in (2). Sometimes, the variant is given inside the form (e.g. *is(t)ność* ‘essence’). In other cases the variant can be given inside the entry description (before a citation which contains such a variant). In cases when a coma is used to separate inflected forms, || signs can be used to separate form variants, e.g. *zachodzca, -e* || *zachojca, -e* (‘deputy’).

In some cases, the format of the entries was invented just for a few atypical cases. Those entries which cannot be parsed using standard rules, will be transfered manually.

5. Inferring part of speech names

The paper dictionary contains only base forms of described words or phrases. The electronic dictionary should contain also their inflected forms. For a highly inflectional language like Polish knowing only base forms is not sufficient not only for automatic text processing. Taking into account change of inflectional patterns in time, it may be difficult for a non-specialist to properly connect an inflected form and the appropriate base form. At the first stage, populating lexicon with wordforms can be done on the basis of citations included in the lexicon – in most cases they include forms which are not base forms of the words they illustrate. These forms should be described with the values of appropriate morphological features which are different for different part of speech (POS) categories. Knowledge of POS category also enables to use information about particular word forms given sometimes within the entry definitions. If, analyzing example (1), we know that the entry describes a noun, we can with a high degree of confidence say that the given ending *-i* describes a genitive from. Unfortunately, the Reczek’s dictionary does not contain this information, so we had to introduce it on the base of other sources.

The idea which was used to decide on the part of speech names for lexicon entries was based on the analysis of given contemporary equivalents (for this task we used morphological analyzer Morfeusz (Saloni et al., 2007)). The entire set of POS names used by the morphological analyzer consists of more than 20 elements, of which most frequent in Polish dictionaries are: *subst* – nouns, *adj* – adjectives, *verb* – verbs, which are represented by their infinitive forms *inf*, *adv* – adverbs, *ger* – gerundium, *ppas* – past participle, *pact* – present participle, *part* – particle and conjunctions *comp* and *conj*. The idea of analyzing contemporary equivalents was supported by the observation that there is a strong tendency to use equivalents which belong to the same grammatical category as the entry they describe. But the situation is not always that simple, and we can distinguish the following types of relations between equivalents and the lexicon entry in the Reczek’s dictionary:

- there is only one single word equivalent given, e.g. palcat «berło» ‘sceptre’
- more that one word equivalents are given, e.g. pal «opał, palenie» ‘fuel, burning’, panownik «władca, panujący» ‘sovereign ruler’
- one of equivalents is a phrase, e.g. parat «figura szermierska», parsk «dół, kopiec ziemny na kartofle, piwnica (za domem), ziemianka, loch»
- the entry itself is a phrase, e.g. {na barzego wsadzić} ‘to encourage’.

All of the distinguished cases had to be covered by the accepted solution. The first case is the most simple one (but not very frequent) and allows for just assigning the entry a POS tag which is assigned to the equivalent (the case of multiple POS tags is treated like the case of multiple POS resulting from the analysis of alternative equivalents). The same strategy can be used if there is more than one equivalent but all of them have the same part of speech assigned. However, in many cases, either equivalents are ambiguous and have more than one POS label or alternative equivalents have different categories. To solve this problem we adopted the voting technique in which all POS labels assigned to all equivalents are given one vote each, with an exception when a base form of a lexeme (interpretation of the equivalent) differs from the equivalent itself, which suggests that the equivalent may be related to a different lexeme; then, POS of the lexeme gains only half of a vote. Votes of all hypothesis are summed up and the lexicon entry is assigned the part of speech which got the highest score. If this procedure does not select one POS label, all names which are most frequently assigned are attributed to the lexicon entry (either the entry itself is ambiguous or the wrong labels will have to be removed at the verification stage). In the first two examples below we got three POS values. Since two of them are equal they point to the part of speech that is finally chosen. In the third example two tags have equal score, so both of them are assigned. For this word both tags are correct, the interpretation can be disambiguated only in the context. The fourth case is an example of the very common ambiguity of nouns and gerunds. This case is often hard to disambiguate even by humans.

- **abecedariusz** «początkujący [pact, subst] 'beginer', nowicjusz [subst] 'novice'» -> [subst]
- **dojutraszek** «kunktator [subst] 'procrastinator', maruda [subst] 'grumbler', jednodniowy [adj] 'one-day'» -> [subst]
- **fuzą** «szybko [adv] 'quickly', migiem [subst]» -> [adv, subst]
- **dufanie** « ufność 'trust' [ger subst])» -> [ger subst]

Additional information may come from the analysis of the entry itself. For the cases when only the meaning of the word changed while its form remained untouched, its morphological description can be known by the morphological analyzer and this vote is also added to the set of results. This strategy is useful in particular for the entries which are added to the lexicon because their occurred on the component list of an entry being a phrase. It could be even more fruitful if some rules of typical orthographic changes would be applied. In the example below, the entry still exists in Polish as a verb with different meaning, while the word used as explanation changed its form.

- **nadziać się** «spodzieć się» (spodziewać się 'to expect') -> inf

For some entries we got no results neither for explanatory words nor the entry itself. In such 174 cases the POS tag remained unknown. In example below, both words used to explain entry are not used (they were replaced by the word *kredowy*).

- **krecisty** «kredowaty, kredzisty» 'chalky'

There are also some cases were the reason of not assigning the POS name lies in the typographic errors of the paper source. In the example below both explanations are typed with errors (although all text was manually verified after the OCR stage, some errors which are due to the OCR itself also remained uncorrected).

- **odgieltować** «odpłaicić, odzwajemnić» (odpłacić 'to pay back' , odwzajemnić 'reciprocate')

In some cases lexicon entries or their equivalents are not one word expressions but phrases (the lexicon contains some idiomatic expressions and sometimes the meaning of the entry is described by means of a phrase). Entries being phrases are assigned their types like the other entries – on the basis of their equivalents, for the latter case we defined a small set of rules to infer a type of phrase on the basis of its constituents.
The adopted ordered set of rules consists of the following heuristics (we neglect words from additional information given in parenthesis like it occurs in the item below):

- ˆ [pos=inf] [orth="się"] => inf, e.g. gładzić się {upiękniać się} "to beautify oneself"
- ˆ [pos=pact] [orth="się"] => subst
- ˆ [pos=pact] [orth! ="się"] => pact
- ˆ [orth="o"] [pos=adj] => adj, e.g. garbonosy$_{adj}$ {o orlim$_{adj}$ nosie$_{subst}$} 'aquiline nose'
- [pos=inf] => inf
- [pos=subst, case=nom] => subst, *hiszpańskie*$_{adj}$ *wino*$_{subst}$ *czerwone*$_{adj}$ *(z*$_{prep}$ *Alcante*$_{subst}$*) 'Spanish red wine from Alcante'*
- [pos=ger, case=nom] => ger, e.g. *wytyczanie*$_{ger,subst}$ *granic*$_{subst}$ 'setting out the borders'

The syntax of the rules allows to specify that a given form has to occur at the beginning of the sequence (ˆ character) or can occur at any place (standard interpretation). A condition can address a particular word form (as 'orth'), a POS name ('pos') or other elements of morphological tags given by a morphological analyzer. In this set of rules only case is addressed. Both equality and non-equality (!=) can be specified. The result is given after the '=>' string.
Below, there is an illustration of the results of applying the rules on two entries for which a two word explanation is given. In the first example the fifth rule which looks for any infinitive verbal form within a sequence is matched. In the second example, the nominal form of a noun ('number') is found.

- **setkować** «karać co setnego» 'punish every hundredth' => inf
- **cetno** «liczba parzysta» 'even number'=> subst

The lexicon contains at the moment 18820 entries. In Table 1 we present the results of the described process of POS assignment. The first part of the table contains numbers of entries which are assigned given number of labels. The first line with the '-' label describes entries which are only references to other entries and are not analyzed (e.g. *odyć zob. 'see' odejść*). There are about 3% of such entries in the Reczek's dictionary. The second line with the '0' label concerns entries which did not get any label as neither they nor

Table 1: POS assignment statistics

number of POS tags	number of entries
-	605
0	174
1	17236
2	888
3	78
4	11

POS lables	number of entries
subst	8828
verb	4718
adj	3051
adv	868
ger	522
ppas	229
pact	141
part	146
comp	62
conj	46

any of their equivalents were recognized by the morphological analyzer we used, or all their equivalents are phrases for which we did not assigned any type. Only less than 1% of entries did not get any label while 92.5% of entries got one POS and 5.2% got more than one POS name assigned. An example of an entry which received 4 tags is given below:

- **spotrzebę** => «dosyć ‘enough’ [imp pred part], wystarczająco [adv]» => [adv impt pred part]

The second part of Table 1 presents frequencies of the most common tags.

The evaluation of the method concerned so far the first 108 entries of the lexicon which were manually assigned POS labels by a qualified linguist. In this set 87 entries were assigned correct POS names, 10 entries were assigned incorrect labels, 4 entries were not assigned any label and there were 7 reference entries which are not analyzed. For the non-referential entries 86% got correct labels. Among the entries which were wrongly described there is one particle which was wrongly qualified as a noun and 2 elements which were assigned only particle labels but should be also described as a subordinative conjunctions (*comp*). The remaining 7 forms were given by the linguists compound labels (a conjunction plus an agglutinative or a conjunction plus a particle) which were treated as separate labels by the tagging program. These compound labels will be introduced in the the second version of the program before all entries will undergo manual verification. However, the results obtained so far already support the claim that assigning POS names on the basis of entries explanations gives reliable results, especially for the most frequent categories like nouns, verbs and adjectives.

Inserting morphological information we assume that all nouns are in nominal singular form (a small set of nouns are given in plural form and descriptions of this set will have to be corrected), and all adjectives are in nominal masculine singular form. To enrich the lexicon with further morphological information we will try to infer the gender of the nouns using a dedicated guesser.

Below, we have the LMF structure representing one entry from the dictionary describing the word *adziamski* ‘Persian’. This word has been assigned two alternative POS names. Which alternative is finally chosen has to be decided by a specialist. This structure has been automatically augmented with the headword’s plural form (*adziamskie*) discovered in the given context (*adziamskie kobierce* ’Persian carpets’).

```
<LexicalEntry id="lex.41">
  <feat att="partOfSpeech" val="subst"
      src="automatic:voting"/>
  <feat att="partOfSpeech" val="adj"
      src="automatic:voting"/>
  <!-- INDEX 40 -->
  <Lemma>
    <FormRepresentation>
      <feat att="writtenForm"
          val="adziamski"/>
      <feat att="language" val="polh"/>
      <feat att="sourceID"
          val="srpsdp:L XVII"/>
    </FormRepresentation>
  </Lemma>
  <WordForm src="automatic:context">
    <FormRepresentation>
      <feat att="writtenForm"
          val="adziamskie"/>
      <feat att="language" val="polh"/>
      <feat att="sourceID"
          val="srpsdp:L XVII"/>
    </FormRepresentation>
  </WordForm>
  <Sense>
    <Definition>
      <TextRepresentation>
        <feat att="writtenForm"
            val="perski"/>     [Persian]
        <feat att="language" val="pol"/>
        <feat att="sourceID" val="srpsdp"/>
      </TextRepresentation>
    </Definition>
    <Context>
      <TextRepresentation>
        <feat att="writtenForm"
            val="Adziamskie kobierce
                 możecie doma zostawić"/>
        [You can leave Persian carpets]
        [at home                      ]
        <feat att="language" val="polh"/>
        <feat att="sourceID"
            val="srpsdp:L XVII"/>
      </TextRepresentation>
    </Context>
  </Sense>
  <RelatedForm targets="lex.8902">
    <FormRepresentation>
      <feat att="writtenForm"
          val="parski"/>
      <feat att="language" val="polh"/>
      <feat att="sourceID" val="srpsdp"/>
    </FormRepresentation>
  </RelatedForm>
</LexicalEntry>
```

6. Summary and further works

Automatic processing of Polish historical texts is still at the very preliminary stage of development, mostly because of the lack of adequate tools and electronic resources. While old text can be (to some extent) accessed in a graphical format of page scans, lexicons which can be easily searched can be of a great importance for the increasing accessibility of historical data. In the paper we presented the process of converting a traditional dictionary of old language variant into an XML format and a method of enriching the constructed dictionary with part of speech names. This step will enable to further extend dictionary with morphological information. The next stage of dictionary development will be to add inflected forms of entries given within citations and to convert information about inflected forms given within original entries by the author into final format. This will require defining heuristics for approximating old declination rules and building forms on the basis of information given by the author of the dictionary. Then, adding more material based on analyzing original texts as well as combining data coming form other dictionaries is planned. The resource is meant to be made available for all old text readers, especially for users of digital libraries whose collections contain more and more old texts.

7. Acknowledgements

This work was supported by SYNAT project financed by the Polish National Centre for Research and Development (SP/I/1/77065/10).

8. References

S. Bąk, M. R. Mayenowa, and F. Pepłowski, editors. 1966-2002. *Słownik polszczyzny XVI wieku, vol. I-XXX.* Zakład Narodowy Imienia Ossolińskich. Wydawnictwo PAN.

F. de Vriend, L. Boves, H. van den Heuvel, and R. van Hout aand J. Swanenberg. 2006. A unified structure for dutch dialect dictionary data. In *Proceedings of The fifth international conference on Language Resources and Evaluation, LREC, Genoa, Italy.*

G. Francopoulo, N. Bel, M. George, M. Calzolari, M. Pet, and C. Soria. 2009. Multilingual resources for NLP in the lexical markup framework (LMF). *Language Resources and Evaluation*, 43:57–70.

A. Mykowiecka, K. Głowińska, P. Rychlik, and J. Waszczuk. 2011. A construction of an electronic dictionary on the base of a paper source. In *Proceedings of Lnaguage and Technology Conference, Poznań.*

Instytut Języka Polskiego PAN. 1996-. *Słownik języka polskiego XVII i 1. połowy XVIII wieku.* http://sxvii.pl.

S. Reczek. 1968. *Podręczny słownik dawnej polszczyzny.* Ossolineum.

L. Romary, S. Salmon-Alt, and G. Francopoulo. 2004. Standards going concrete: from LMF to Morphalou. In *Proceedings of the 20th International Conference on Computational Linguistics, COLING 2004, Geneve.*

Z. Saloni, W. Gruszczyński, M. Woliński, and R. Wołosz. 2007. *Słownik gramatyczny języka polskiego.* Wiedza Powszechna.

The TXM Portal Software giving access to Old French Manuscripts Online

Serge Heiden, Alexei Lavrentiev
ICAR Research Lab – Lyon University and CNRS
ENS de Lyon
15 parvis René Descartes
69007 Lyon
E-mail: slh@ens-lyon.fr, alexei.lavrentev@ens-lyon.fr

Abstract

This paper presents the new TXM software platform giving online access to Old French Text Manuscripts images and tagged transcriptions for concordancing and text mining. This platform is able to import medieval sources encoded in XML according to the TEI Guidelines for linking manuscript images to transcriptions, encode several levels of diplomatic transcription including abbreviations and word level corrections. It includes a sophisticated tokenizer able to deal with TEI tags at different levels of linguistic hierarchy. Words are tagged on the fly during the import process using IMS TreeTagger tool with a specific language model. Synoptic editions displaying side-by-side manuscript images and text transcriptions are automatically produced during the import process. Texts are organized in a corpus with their own metadata (title, author, date, genre, etc.) and several word property indexes are produced for the CQP search engine to allow efficient word patterns search to build different types of frequency lists or concordances. For syntactically annotated texts, special indexes are produced for the Tiger Search engine to allow efficient syntactic concordance building. The platform has also been tested on classical Latin, ancient Greek, Old Slavonic and Old Hieroglyphic Egyptian corpora (including various types of encoding and annotations).

Keywords: Old French, textometry, TEI, tokenizer, synoptic edition

1. The TXM platform

Textometry, born in France in the 1980's, has developed powerful techniques for the analysis of large bodies of texts. Following lexicometry and text statistical analysis, it offers tools and statistically well founded methods tested in multiple branches of the humanities.

The TXM platform combines powerful techniques for the analysis of large bodies of texts in a modular and open-source framework (Heiden, 2010; Heiden et al., 2010; Pincemin et al., 2010). It was initiated by the Textométrie research project[1] funded by the French ANR research council (2007-2010) which brought together various previous Textometry software developments. It is both a heritage of international influence and the launch of a new generation of textometrical research, in synergy with existing corpus technologies (Unicode, XML, TEI, NLP). The development of the software continues thanks to funding from various research projects and especially in the framework of the Matrice Equipex research infrastructure funded by the French government (2012-2021).

The platform is delivered in the form of two applications: a standalone version for local computer systems (Linux, Windows and Mac OS) and a GWT-based[2] web portal application for online corpus analysis. The source code and application setups can be downloaded for free from Sourceforge.net[3] under a GPL v3 licence.

The current version includes the following features:

- building a subcorpus based on any metadata (date, author, genre, etc.) at any structural level (text, section, etc.) of a corpus;
- KWIC concordance building of word pattern queries (based on the CQP search engine[4]);
- paginated editions or 'text view' of all the texts of a corpus;
- frequency lists of word pattern queries;
- various statistical analysis based on R[5] packages (factorial correspondance analysis, analysis of cooccurring words or lexical patterns, etc.)
- importing from various textual sources to build corpora. Available import modules include raw text combined to flat metadata (CSV), raw XML/w+metadata, XML-TEI BFM[6] , XML-TXM[7], XML-Transcriber+metadata, etc.;
- running NLP tools on the input files during the import process. Currently three different tokenizers are available (one of which is TEI compatible) and TreeTagger and TnT plugins are available for POS tagging. Annotations are then available inside the platform as word properties usable with the search engine;
- exporting all results in CSV for lists and tables or in SVG for graphics;
- R and Groovy scripting.

[1] http://textometrie.ens-lyon.fr/?lang=en

[2] http://code.google.com/intl/en/webtoolkit

[3] http://sourceforge.net/projects/txm

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

[5] http://www.r-project.org

[6] As defined by the TEI compatible XML text encoding guidelines of the BFM project (http://bfm.ens-lyon.fr/article.php3?id_article=158).

[7] An XML-TEI extension NLP oriented TXM internal pivot format (http://sourceforge.net/apps/mediawiki/txm/index.php ?title=Xml-txm-tei).

The whole import and NLP annotation process environment are implemented in the standalone version. Resulting ‘binary’ corpora can be uploaded to a web portal version and then searched online publicly or with access control restrictions.

The TXM platform is currently used in research projects in various fields of the humanities, such as history, literature, geography, linguistics and political science.

In this paper we will focus on the way TXM has been adapted to deal with TEI-encoded synoptic editions of Old French language texts including rich editorial markup in their digital transcription.

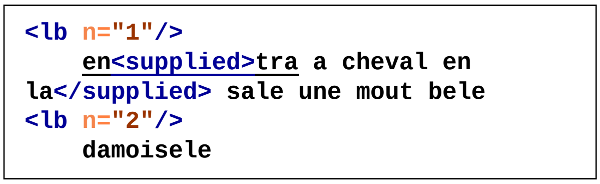
The closest integrated corpus analysis platform to TXM combining the four objectives of XML TEI import, full text search engine, statistical analysis and graphical user interface is Philologic[8] associated to Philomine[9]. However, Philologic only reads TEI lite encoding, doesn’t help with NLP tools, provides less statistical analysis tools and is only available through a web based user interface. GATE[10] and UIMA[11] are general frameworks available for corpus NLP annotation but are not specialized in XML encoding, see (Heiden, 2010) for a discussion, and don't provide an integrated user interface for online corpus analysis and statistical tools. CQPweb[12] provides a web user interface to the same full text search engine as TXM, but doesn't provide services for XML TEI sources import, NLP tools or statistical analysis.

2. Analyzing Old French TEI XML encoded sources with TXM

2.1 Tokenizing and tagging sentences in the import process

One of the greatest challenges in applying NLP annotation and corpus query tools to texts with rich philological markup is to identify correctly the words and the sentences to which the linguistic annotation will apply without losing relevant editorial information.

As an example, the following XML encoding of a transcription is perfectly legal in TEI but it makes the job of a tokenizer very hard as the editorial correction starts in the middle of a word and continues over a few following words:



```
<lb n="1"/>
    en<supplied>tra a cheval en
la</supplied> sale une mout bele
<lb n="2"/>
    damoisele
```

Figure 1: Example of TEI philological markup overlapping with linguistic structure before tokenization

Even more problems arise when it comes to the markup of sentences, especially in verse texts where line and sentence structure often overlap.

It is of course possible to “filter out” all tags that can overlap with basic linguistic structures (words and sentences) but this may result in loss of relevant data for query formulation and rendering (e.g. whether a word comes from the original document or it has been corrected by the editor).

It appears to be virtually impossible to work out a tokenization algorithm that would process correctly any TEI encoded text containing rich philological markup. However good results can be obtained if the source documents comply with a number of basic rules, such as: “tags that appear inside a word should be clearly identified” or “if a text span starts inside a word and continues over the next words, it should be split”. It is also possible to establish and customize lists of TEI tags according to their position in the linguistic hierarchy in a given encoding project. For instance, block-level elements, like <p> or <ab> are necessarily superior to sentences. Some tags apply to text spans of one or several words within a sentence (e.g. <name>, <date>, <ref>). In a given project the list of tags with such a behavior can be extended (e.g. assume that a <foreign> is never superior to a sentence or that a <corr> applies at least to one whole word). Some tags may be considered to be word-level (e.g. <abbr>, <num>, <pc>) and a small number of tags is most often word-internal (<am>, <c>, <ex>). It is also possible to identify the elements containing text spans that should not be tokenized as they do not belong to the original source (e.g. editor’s notes in a critical edition). Once the tag classes are clearly defined, the number of elements that are likely to overlap with the linguistic hierarchy can be considerably reduced.

At present, the most sophisticated tokenization method within the TXM import process has been elaborated for the TEI-encoded texts of the Base de Français Médiéval (BFM). The BFM project applies to its texts a clearly defined and documented TEI ODD customization schema (Heiden, Guillot et al., 2010) that includes rules on dealing with word-internal tags (such as missing letters supplied by the editor or hyphenation marks[13]) and elements that are likely to overlap with sentences (e.g. empty <lb/> tags are used instead of <l> elements to mark up verse lines, quotation <q> tags are considered to be “sentence-breaking”, etc.). The BFM import module, including the tokenizer presented here[14], is available for all users of TXM standalone version.

A few tests on adapting the BFM tokenizer to the XML TEI documents produced by external projects and using different customization schemes have been recently

[8] http://sites.google.com/site/philologic3

[9] http://code.google.com/p/philomine

[10] http://gate.ac.uk

[11] http://uima.apache.org

[12] http://www.ling.lancs.ac.uk/activities/713

[13] According to the BFM encoding rules, words containing this kind of tags should be pre-tokenized using the <w> tag

[14] Currently, using this module on a corpus of texts that are not conformant with BFM encoding rules may result in aborting the import process. A more robust algorithm is being designed.

performed. These include texts from the 16th century French BVH virtual library[15], the online edition of Flaubert's *Bouvard et Pécuchet* preparatory files[16] and Frantext corpus[17] texts. The adaptation procedure consists in applying specific XSL filters before and after the tokenization and in adjusting the tag classes. The pre-tokenization filter leaves out the tags that are not relevant for use with TXM, and simplifies and normalizes certain complex XML structures. The post tokenization filter corrects some issues that are very hard to deal with during the primary tokenization (e.g. page-end hyphenation where a considerable number of tags can separate the beginning and the end of a word). All XSL filters for pre- and post-tokenization processing (filter-*.xsl) are included in the TXM standalone version[18]. The TXM standalone version local interface allows applying an arbitrary XSLT2 stylesheet and customizing TEI tag classes prior to import process. To apply a "post-tokenization" filter, it is necessary to copy the tokenized files from the dedicated folder in the TXM corpus workspace and to re-import them to TXM applying the relevant post-tokenization XSL

The tokenization process produces an XML-TXM (TXM specific TEI extension pivotal XML format) formatted file where every word relevant for corpus queries is marked up with a <w> tag bearing a unique identifier within the corpus. The TEI extension consists in redefining the content model of <w> which includes one ore more <txm:form> elements representing the default and alternative (optional) word-forms (e.g. a normalized form with the distinction of *u* and *v* and a "diplomatic" form that follows the usage of the original manuscript) and zero or more <txm:ana> elements that contain all kinds of word-level annotation (*cf.* Figure 2). Those extensions are a minimal design with the sole purpose of [19]:

- repeating the various graphical forms encoding (in <txm:form> sub-elements);
- repeating the various word level NLP annotations (in <txm:ana> sub-elements) and being able to link them to a person responsible or an annotation tool and its calling parameters, declared in the teiHeader).

Sentences are optionally marked up using <s> tags.

Relevant data from XML-TXM files can be exported in appropriate formats for various NLP tools (e.g. the TreeTagger for part-of-speech tagging and lemmatization) as well as for (semi-)manual annotation (e.g. a spreadsheet for the verification of automatic part-of speech tagging). At the following processing stage, the annotation data is "injected" into the XML-TXM file in the form of additional <txm:ana> elements.

```
<w type="NOMpro" xml:id="w_fro_000100">
  <txm:form
    type="norm">Lancelot
 </txm:form>
  <txm:form
    type="dipl">lanc<ex>elot</ex>
 </txm:form>
  <txm:form
    type="facs">lanc̄.
 </txm:form>
  <txm:ana type="pos">NOMpro</txm:ana>
  <txm:ana type="q">1</txm:ana>
</w>
```

Figure 2: Example of a word token in XML-TXM format after tokenization and tagging

Some additional details on various TXM tokenization methods can be found in (Heiden, 2010).

2.2 CATTEX morpho-syntactic tagging and querying

Like any language without normative grammar and with a large area of dissemination, Old French is characterized by a great variability of spelling, morphology and vocabulary. An additional complexity factor is created by the diachronic evolution of texts in a corpus like the BFM covering nearly 500 years of the language history: it may be hard to determine at what point a certain category or analysis is no longer applicable. The absence of "native speakers" makes it impossible to produce new experimental data. Therefore, it is a real challenge to create NLP tools for effective morpho-syntactic tagging and lemmatization of old texts.

The BFM project made the choice of elaborating a specific tagset for morpho-syntactic description of Old and Middle French texts called CATTEX and to produce detailed guidelines for human annotators (Prévost et al. 2011). These guidelines are constantly revised and completed as new texts are being processed. A few texts tagged manually at the first stage of the project were used as a training corpus for several NLP tools, and in particular the TreeTagger.

Since 2010 all the texts of the BFM are processed by TreeTagger during the import process using the language model (or parameter file) based on previously tagged texts. Some of the texts are subsequently verified and corrected by human experts and are progressively included in the "gold standard" training corpus that is used to produce the language model for TreeTagger[20]. The results of automatic tagging are considerably improved as the training corpus grows. In the future, it is planned to produce more specific training corpora based on multi-dimensional text variation analysis (date of

[15] http://www.bvh.univ-tours.fr

[16] http://dossiers-flaubert.ish-lyon.cnrs.fr

[17] http://www.frantext.fr

[18] And also directly from TXM source repository: http://txm.svn.sourceforge.net/viewvc/txm/trunk/Toolbox/trunk /org.textometrie.toolbox/res/org/txm/xml/xsl

[19] Using TEI feature structures <fs> for more complex analytic annotations is being considered for future developments.

[20] The Old French CATTEX TreeTagger language model is available at http://bfm.ens-lyon.fr/article.php3?id_article=324

creation, dialectal features, text domain and genre).

Tagged corpora can then be queried by the very efficient CQP search engine included in the TXM platform. As an illustration, figure 3 shows a sample display of a frequency list of all the different patterns of the following CQL query in one text of the BFM:

```
 [pos="ADJ.*"] [word="chevaliers?"%c]
```

The sequence of square brackets `[…] […]` express a sequence of two words, of which, the first one must obey the `pos=”ADJ.*”` constraint (i.e. its morpho-syntactic tag begins with “ADJ”) and the second must obey the `word=”chevaliers?”%c` constraint (i.e. it has the graphical form “chevalier” or “chevaliers”) in any character case (“`%c`” modifier).

2.3 SRCMF Syntactic annotation querying

Some of the BFM texts have been annotated for syntactic relations in the framework of the SRCMF[21] project using a specially designed dependency-based linguistic model. The annotation was performed and cross-checked by human experts using the NotaBene open source RDF annotation tool (Mazziotta, 2010)[22]. Annotated texts can be exported to the Tiger-XML format (Lezius, 2002) and then be queried by the TigerSearch engine included in the TXM platform. As an illustration, figure 4 shows the second matching tree of the following TigerSearch query in the same text as previously:

```
[word="bon"] . #c:[word="chevalier"] &
[cat="Cmpl"] >L #c
```

The square brackets part of the query has similar semantics to the CQP syntax but the dot (`.`) between them express here precedence. The second terminal (*chevalier*) is labeled “`#c`” and further constrainted to be directly dominated by a complement node (*Cmpl*) through an *L* edge (the *chevalier* word must be a complement in the syntactic structure).

2.4 Building Synoptic Editions Including Manuscript Images

TXM builds paginated editions of the source texts for reading and browsing. If the source text markup includes multiple readings or presentation forms (e.g. more or less “diplomatic” transcription), it is possible to produce multiple versions (or facets) of the edition. It is also possible to link images of the original manuscript or book to every page or column of the edition. Any combination of these can be displayed side by side using the TXM “Edition” functionality. In other words, TXM allows building highly customizable editions of the source texts.

The TXM platform provides a high quality on-screen edition of the texts of a corpus that are linked from KWIC concordances. A double-click on a line of a concordance opens an edition view of the page containing the selected occurrence (Figure 5). The keyword occurrence is highlighted with a red background. Hovering over any word of an edition with the mouse displays a flyover of its morpho-syntactic description ('NOMcom' - common noun - CATTEX tag in the *chevalier* example) and other token-level annotations available for the corpus.

2.5 Importing and using text metadata

Metadata are extremely important for corpus analysis, and the TXM platform offers a number of tools for creating and analysing subcorpora and corpus “partitions” based on metadata associated with whole texts or with text-internal structural units.

In the BFM, the metadata adapted to Old French manuscripts are stored in an external relational database. Before the texts are imported to TXM, the relevant metadata are exported to the TEI header of each file. This operation ensures the integrity of each text with its metadata and facilitates data exchange with research partners.

The metadata available for BFM texts on the TXM platform include the name of the author, text title, date of creation of the manuscript, author’s dialect, text form (prose, verse or mixed), domain, genre, etc. Figure 6 illustrates the sub-corpus building high-level interface of the TXM platform in which the user can see in real time histogram word statistics of the sub-corpus being built by metadata values selection.

2.6 Hosting the BFM corpus online in a TXM portal

The BFM corpus can be freely accessed online through a TXM portal at http://txm.bfm-corpus.org/bfm[23], after having subscribed to the portal and accepted the BFM access conditions charter. Once connected, the user can use all the available TXM analysis tools (KWIC concordances, frequency lists, specificity statistics, etc.) with the following restrictions: some texts are not allowed to be read in an online edition and the size of the contexts of KWIC concordances may be limited. The XML-TEI sources of some texts, like the “Queste del saint Graal” Old French edition, are available for download under a Creative Commons BY-NC licence through the portal.

3. Conclusion

The TXM platform, which was initially designed for modern language corpora, has been successfully adapted to help analyze various ancient language corpora and will be developed further. The future versions of the platform will be able to analyze parallel corpora and to help researchers annotate collaboratively texts while analyzing them online.

References

[21] Syntactic Reference Corpus of Medieval French, project funded by ANR-DFG joint French-German program (2009-2012), resp. Sophie Prévost (Lattice research laboratory) and Achim Stein (Stuttgart University), http://srcmf.org (under construction).

[22] The software is freely available at http://sourceforge.net/projects/notabene.

[23] The portal should be opened to the public in April 2012

Christ, O. (1994). A Modular and Flexible Architecture for an Integrated Corpus Query System. In Proc. of COMPLEX'94 (3rd Conf. on Computational Lexicography and Text Research), pp. 23-32.

Pincemin, B., Heiden, S., Lay, M.-H., Leblanc J.-M. and Viprey, J.-M. (2010) Fonctionnalités textométriques: Proposition de typologie selon un point de vue utilisateur. In S. Bolasco et al. (Eds.), *Statistical Analysis of Textual Data - Proceedings of 10th International Conference JADT 2010*, Edizioni Universitarie di Lettere Economia Diritto, Rome, 9-11 juin 2010.

Heiden, Serge (2010) The TXM Platform: Building Open-Source Textual Analysis Software Compatible with the TEI Encoding Scheme. 24th Pacific Asia Conference on Language, Information and Computation. Éd. Kiyoshi Ishikawa Ryo Otoguro. Institute for Digital Enhancement of Cognitive Development, Waseda University, 2010. 389-398. online.

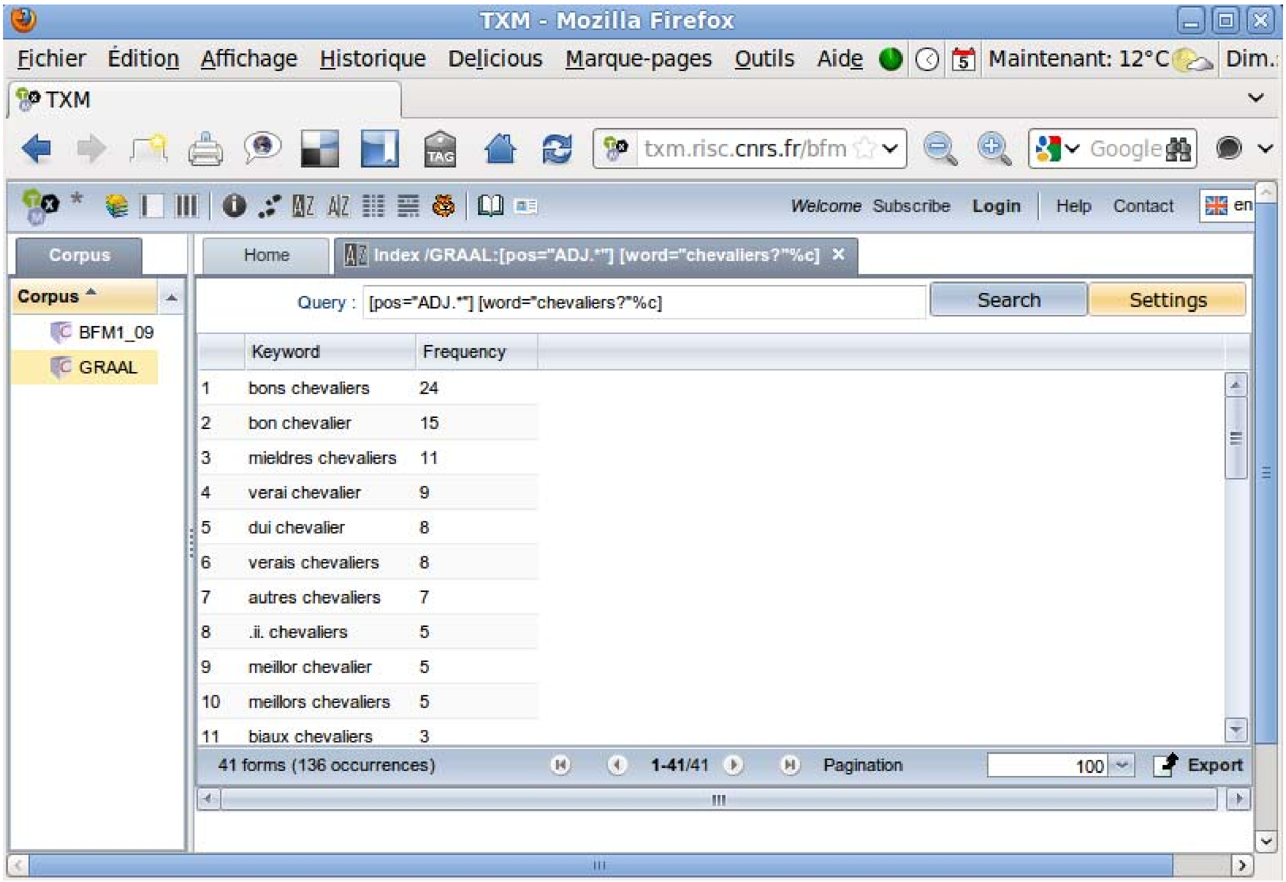
Heiden Serge, Magué Jean-Philippe, Pincemin Bénédicte (2010) TXM : Une plateforme logicielle open-source pour la textométrie – conception et développement, in Sergio Bolasco&al (eds), Statistical Analysis of Textual Data -Proceedings of 10th International Conference JADT 2010.

Heiden, S., Guillot, C., Lavrentiev, A., Bertrand, L. (2010). *Manuel d'encodage XML-TEI des textes de la Base de Français Médiéval*, Lyon, Équipe BFM

Lezius, W. (2002).TIGERSearch – Ein Suchwerkzeug für Baumbanken // Proceedings der 6. Konferenz zur Verarbeitung natürlicher Sprache (KONVENS 2002). Saarbrücken. 2002. http://konvens2002.dfki.de.

Mazziotta, N. (2010). Building the Syntactic Reference Corpus of Medieval French using NotaBene RDF Annotation Tool. Proceedings of the Fourth Linguistic Annotation Workshop. Stroudsburg, PA, USA. P. 142–146. http://sourceforge.net/projects/notabene

Prévost, S., Guillot, C., Lavrentiev, A., Heiden, S. (2010). *Jeu d'étiquettes CATTEX2009*, version 1.3. Lyon, Projet BFM, http://bfm.ens-lyon.fr/IMG/pdf/Cattex2009_1.3.pdf

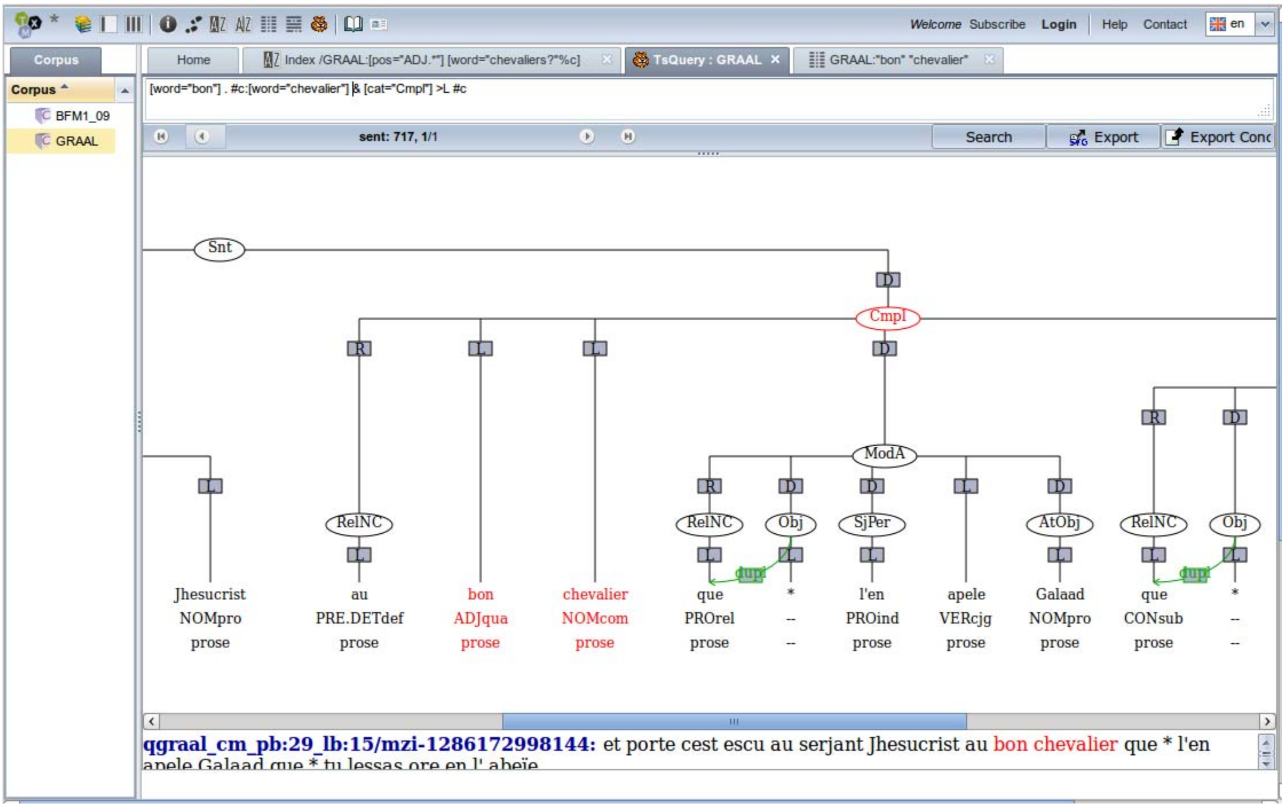

TXM - Mozilla Firefox
Fichier Édition Affichage Historique Delicious Marque-pages Outils Aide
Maintenant: 12°C
Dim.:
TXM
txm.risc.cnrs.fr/bfm
Google
Welcome Subscribe Login Help Contact
en
Corpus
Home
Index /GRAAL:[pos="ADJ.*"] [word="chevaliers?"%c]
Corpus
BFM1_09
GRAAL
Search
Settings
Pagination
Export

Query : [pos="ADJ.*"] [word="chevaliers?"%c]

	Keyword	Frequency
1	bons chevaliers	24
2	bon chevalier	15
3	mieldres chevaliers	11
4	verai chevalier	9
5	dui chevalier	8
6	verais chevaliers	8
7	autres chevaliers	7
8	.ii. chevaliers	5
9	meillor chevalier	5
10	meillors chevaliers	5
11	biaux chevaliers	3

41 forms (136 occurrences) 1-41/41 Pagination 100

Figure 3: TXM portal session in the Firefox browser showing a frequency list of all sequences of adjectives followed by *chevalier* (some of them in plural form) in the "Queste del Saint Graal" text from the BFM.



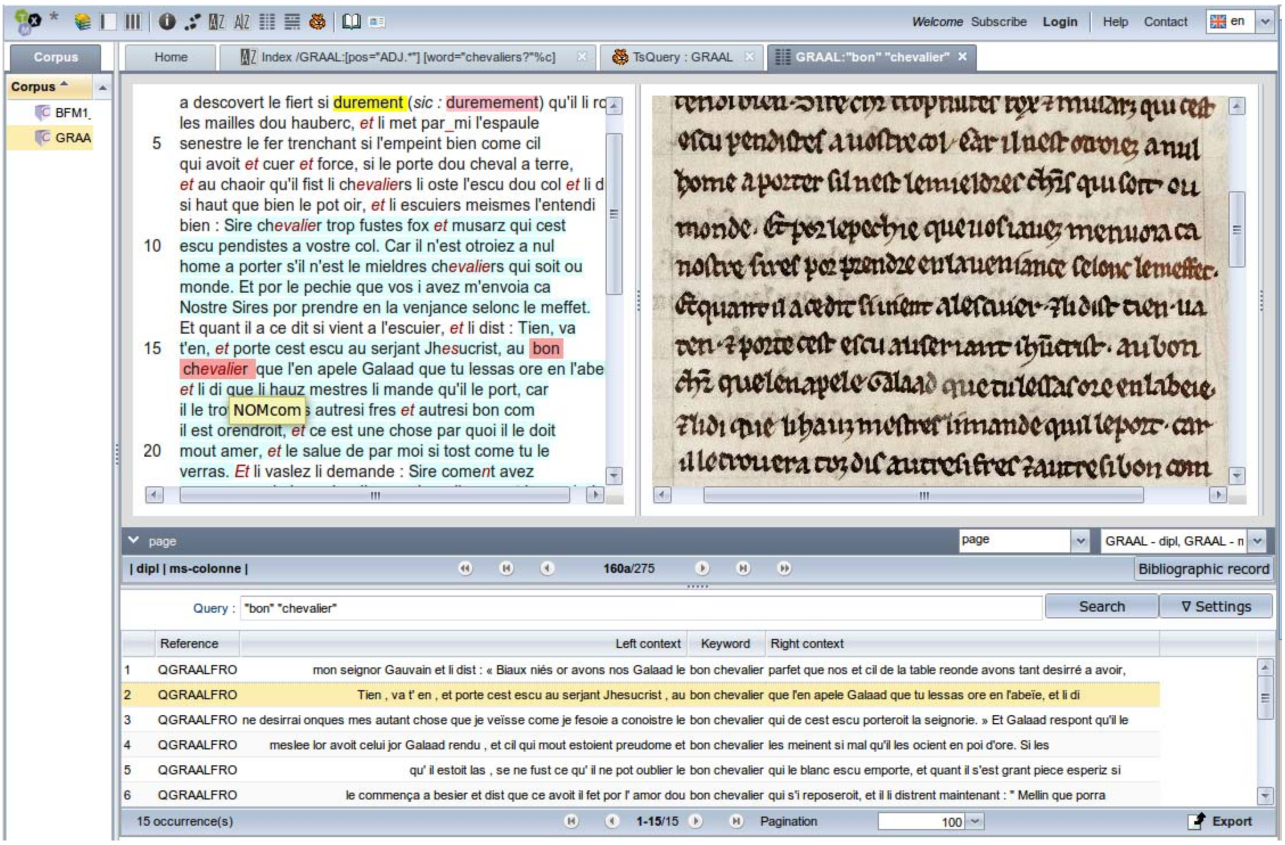
[word="bon"] . #c:[word="chevalier"] & [cat="Cmpl"] >L #c

sent: 717, 1/1

Jhesucrist	au	bon	chevalier	que	*	l'en	apele	Galaad	que	*
NOMpro	PRE.DETdef	ADJqua	NOMcom	PROrel	--	PROind	VERcjg	NOMpro	CONsub	--
prose	prose	prose	prose	prose	--	prose	prose	prose	prose	--

qgraal_cm_pb:29_lb:15/mzi-1286172998144: et porte cest escu au serjant Jhesucrist au bon chevalier que * l'en apele Galaad que * tu lessas ore en l' abeïe

Figure 4: Display in the TXM portal of a syntactic tree matching the *bon chevalier* words sequence in which *chevalier* is a complement in the “Queste del Saint Graal” text from the BFM.



a descovert le fiert si durement (*sic :* duremement) qu'il li ro
les mailles dou hauberc, *et* li met par_mi l'espaule
senestre le fer trenchant si l'empeint bien come cil
qui avoit *et* cuer *et* force, si le porte dou cheval a terre,
et au chaoir qu'il fist li chevaliers li oste l'escu dou col *et* li d
si haut que bien le pot oir, *et* li escuiers meismes l'entendi
bien : Sire chevalier trop fustes fox *et* musarz qui cest
escu pendistes a vostre col. Car il n'est otroiez a nul
home a porter s'il n'est le mieldres chevaliers qui soit ou
monde. Et por le pechie que vos i avez m'envoia ca
Nostre Sires por prendre en la venjance selonc le meffet.
Et quant il a ce dit si vient a l'escuier, *et* li dist : Tien, va
t'en, *et* porte cest escu au serjant Jhesucrist, au bon
chevalier que l'en apele Galaad que tu lessas ore en l'abe
et li di que li hauz mestres li mande qu'il le port, car
il le tro s autresi fres *et* autresi bon com
il est orendroit, *et* ce est une chose par quoi il le doit
mout amer, *et* le salue de par moi si tost come tu le
verras. *Et* li vaslez li demande : Sire coment avez

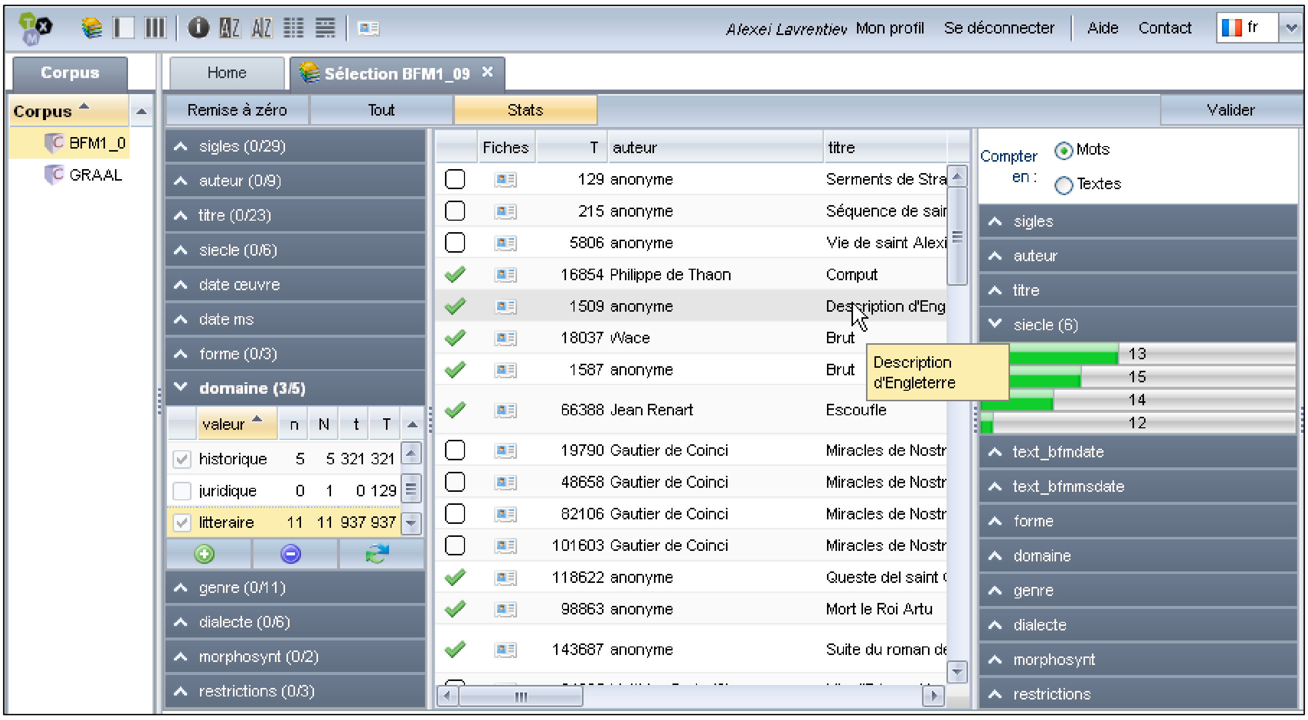
Query : "bon" "chevalier"

	Reference	Left context	Keyword	Right context
1	QGRAALFRO	mon seignor Gauvain et li dist : « Biaux niés or avons nos Galaad le	bon chevalier	parfet que nos et cil de la table reonde avons tant desirré a avoir,
2	QGRAALFRO	Tien , va t' en , et porte cest escu au serjant Jhesucrist , au	bon chevalier	que l'en apele Galaad que tu lessas ore en l'abeïe, et li di
3	QGRAALFRO	ne desirrai onques mes autant chose que je veïsse come je fesoie a conoistre le	bon chevalier	qui de cest escu porteroit la seignorie. » Et Galaad respont qu'il le
4	QGRAALFRO	meslee lor avoit celui jor Galaad rendu , et cil qui mout estoient preudome et	bon chevalier	les meinent si mal qu'il les ocient en poi d'ore. Si les
5	QGRAALFRO	qu' il estoit las , se ne fust ce qu' il ne pot oublier le	bon chevalier	qui le blanc escu emporte, et quant il s'est grant piece esperiz si
6	QGRAALFRO	le commença a besier et dist que ce avoit il fet por l' amor dou	bon chevalier	qui s'i reposeroit, et il li distrent maintenant : " Mellin que porra

15 occurrence(s) 1-15/15

Figure 5: Synoptic edition in the TXM portal of folio *160* column *a* of the “Queste del Saint Graal” text from the BFM upon a KWIC concordance of the *bon chevalier* pattern built with CQP. The edition was opened by a double-click on

the second hit of the concordance (same matching words as in the previous syntactic example). The left side is the diplomatic version of the transcription (quotations or direct speech are rendered in blue background, editorial markup in pink and yellow, etc.) and the right side is the facsimile image.



valeur	n	N	t	T
☑ historique	5	5	321	321
□ juridique	0	1	0	129
☑ litteraire	11	11	937	937

	Fiches	T	auteur	titre
□		129	anonyme	Serments de Stra
□		215	anonyme	Séquence de sair
□		5806	anonyme	Vie de saint Alexi
✓		16854	Philippe de Thaon	Comput
✓		1509	anonyme	Description d'Eng
✓		18037	Wace	Brut
✓		1587	anonyme	Brut
✓		66388	Jean Renart	Escoufle
□		19790	Gautier de Coinci	Miracles de Nostr
□		48658	Gautier de Coinci	Miracles de Nostr
□		82106	Gautier de Coinci	Miracles de Nostr
□		101603	Gautier de Coinci	Miracles de Nostr
✓		118622	anonyme	Queste del saint (
✓		98863	anonyme	Mort le Roi Artu
✓		143687	anonyme	Suite du roman de

Compter en : ● Mots ○ Textes

Figure 6: Sub-corpus building in the BFM corpus with the TXM portal Text Metadata Selection Interface. In the left panel, the "historique" and "littéraire" DOMAIN (domaine) metadata values are selected, the corresponding texts are checked in the middle panel and the resulting number of words for each CENTURY (siècle) metadata value is displayed in real time as a green histogram in the right panel for diachronic balance judgment.

Making Historical Latvian Texts More Intelligible to Contemporary Readers

Lauma Pretkalniņa, Pēteris Paikens, Normunds Grūzītis, Laura Rituma, Andrejs Spektors
Institute of Mathematics and Computer Science, University of Latvia
Raiņa blvd. 29, LV-1459, Riga, Latvia
E-mail: lauma@ailab.lv, peteris@ailab.lv, normundsg@ailab.lv, laura@ailab.lv, aspekt@ailab.lv

Abstract

In this paper we describe an ongoing work developing a system (a set of web-services) for transliterating the Gothic-based Fraktur script of historical Latvian to the Latin-based script of contemporary Latvian. Currently the system consists of two main components: a generic transliteration engine that can be customized with alternative sets of rules, and a wide coverage explanatory dictionary of Latvian. The transliteration service also deals with correction of typical OCR errors and uses a morphological analyzer of contemporary Latvian to acquire lemmas – potential headwords in the dictionary. The system is being developed for the National Library of Latvia in order to support advanced reading aids in the web-interfaces of their digital collections.

1. Introduction

In 2010, a mass digitalization of books and periodicals published from the 18th century to the year 2008 was started at the National Library of Latvia (Zogla and Skilters, 2010). This has created a valuable language resource that needs to be properly processed in order to achieve its full potential and accessibility to a wide audience, especially in the case of historical texts.

A fundamental issue in a massive digitalization of historical texts is the optical character recognition (OCR) accuracy that affects all the further processing steps. The experience of Tanner et al. (2009) shows that only about 70–80% of correctly recognized words can be expected in the case of the 19th century English newspapers. The actual OCR accuracy achieved in the digitalization of the National Library of Latvia (NLL) corpus has not been systematically evaluated yet[1], however, in the case of historical Latvian, at least two more obstacles have to be taken into account: the Gothic-based Fraktur script (that differs from the Fraktur used in historical German) in contrast to the Latin-based script that is used nowadays, and the inconsistent use of graphemes over time.

During the first half of the 20th century, the Latvian orthography has undergone major changes and has acquired its current form only in 1957[2]. The Fraktur script used in texts printed as late as 1936 is not familiar to most readers of contemporary generation. Moreover, the same phonemes are often represented by different graphemes, even among different publishers of the same period. The Latvian lexicon, of course, has also changed over time, and many words are not widely used and known anymore.

This makes a substantial obstacle in the accessibility of Latvian cultural heritage, as almost all pre-1940 printed texts currently are not accessible to contemporary readers in an easily intelligible form.

In this paper we describe a recently developed system for transliterating and explaining tokens (on a user request) in various types of historical Latvian texts.

In the following chapters, we first give a brief introduction to the evolution of the Latvian orthography, and then we describe the design and implementation of the system that aims to eliminate the accessibility issues (to a certain extent). We also illustrate some use-cases that hopefully will facilitate the use of the Latvian cultural heritage.

2. Latvian orthography

The first printed works in Latvian appeared in the 16th century. Until the 18th century the spelling was highly inconsistent, differing for each printed work. Since the 18th century a set of relatively stable principles has emerged, based on the German orthography adapted to represent the Latvian phonetic features (Ozols, 1965).

In 1870-ies, with the rise of national identity, there were first activities to develop a new orthography that would be more appropriate to describe the sounds used in Latvian: long vowels, diphthongs, affricates, fricatives and palatalized consonants (Paegle, 2001). This goes hand in hand with the slow migration from the Fraktur script to the Latin script. The ultimate result of these efforts was an alphabet that in almost all cases has a convenient one-to-one mapping between letters and phonemes, and is almost the same as the modern Latvian alphabet that consists of 33 letters. However, the adoption of these changes was slow and inconsistent, and both scripts were used in parallel for a prolonged time (Paegle, 2008). From around 1923, Latvian books are mostly printed in the Latin script, but many newspapers still kept using the Fraktur script until late 1930-ies due to investments in the printing equipment.

There were additional changes introduced in the modern orthography in 1950-ies, eliminating the use of graphemes ‘ch’ and ‘ŗ’, and changing the spelling of many foreign words to imitate their pronunciation in Russian. This once again resulted in decades of parallel orthographies: texts printed in USSR use the new spelling while texts published in exile resist these changes.

This presents a great challenge, as the major orthographic changes have occurred relatively late and, thus, a huge proportion of Latvian printed texts have been published in obsolete orthographies. Furthermore, the

[1] The expected accuracy is about 80% at the letter level.
[2] http://en.wikipedia.org/wiki/Latvian_language#Orthography

available linguistic resources and tools, such as dictionaries and morphological analyzers, do not support the historical Latvian orthography.

Figure 1 illustrates some of the issues that have to be faced in the processing pipeline if one would semi-automatically convert a text in Fraktur into the modern Latvian orthography. It should be mentioned that, in the scope of this project, OCR is provided by a custom edition of ABBYY FineReader (Zogla and Skilters, 2010).

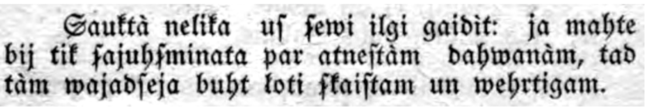
The original facsimile (the old Fraktur orthography):
 Sauktà nelika uſ ſewi ilgi gaidit: ja mahte bij tik ſajuhſminata par atneſtàm dahwanàm, tad tàm wajadſeja buht łoti ſkaiſtam un wehrtigam.
The actual result of OCR:
Sauktà nelika us sewi ilgi gaidît: ja mahte bij tik sajuhsminata par atnestām dahroanàm, tad tàm roajadseja buht ļoti skaistam un wehrtigam.
The expected OCR result (Latin script, old orthography):
Sauktā nelika uz sewi ilgi gaidīt: ja mahte bij tik sajuhsminata par atnestām dahwanām, tad tām wajadzeja buht ļoti skaistam un wehrtigam.
Transliteration into the modern orthography:
Sauktā nelika uz sevi ilgi gaidīt: ja māte bija tik sajūsmināta par atnestām dāvanām, tad tām vajadzēja būt ļoti skaistām un vērtīgām.

Figure 1: A sample sentence in the historical Latvian orthography and its counterpart in the modern orthography along with intermediate representations.

3. Transliteration engine

We have developed a rule-based engine for performing transliterations and correcting common OCR errors. In this chapter we describe the engine assuming that rules defining the transliteration and error correction are already provided.

To satisfy the user interface requirements[3], the engine is designed to process a single token at a time. The workflow can be described as follows:

- The input data is a single word (in general, an inflected form).
- Find all transliteration rules that might be applied to the given word and apply them in all the possible combinations (thus acquiring potentially exponential amount of variants).
- Find the potential lemmas for the transliteration variants using a morphological analyzer of the contemporary language (Paikens, 2007).
- Verify the obtained lemmas against large, authoritative wordlists containing valid Latvian words (in the modern orthography) of various domains and styles, as well as of regional and historical lexicons.
- Assign a credibility level to each of the proposed variants according to the transliteration and validation results. In an optional step, the transliteration variants (both wordforms and lemmas) can be ranked according to their frequency in a text corpus.

Note that the contextual disambiguation of the final variants (if more than one) is left to the reader.

Below we shall describe most significant parts of the workflow in more depth.

3.1 Types of transliteration rules

Our transliteration engine uses two types of rules: obligatory and optional. The obligatory rules describe reliable patterns (usually for the standard transliteration, but also for common OCR error correction) that are always applied to the given word, assuming that in practice they will produce mistakes only in rare cases. When this set of rules is applied to a target string, only one replacement string is returned (except cases when a target string is a substring[4] of another target string; see Figure 2: 'tsch' vs. 'sch').

The optional rules describe less reliable patterns (usually for OCR correction, but also for transliteration) that should be applied often, but not always. I.e., the optional rules produce additional variants apart from the imposed ones (by the obligatory rules). When a set of optional rules is applied, it is allowed to return more than one replacement string for a given target string.

All rules are applied "simultaneously", and the same target string can be matched by both types of rules (e.g. a standard transliteration rule is that the letter 'w' is replaced by 'v', however, the Fraktur letter 'm' is often mistakenly recognized as 'w').

Figure 2 illustrates various rules of both types (some of them are applied to acquire the final transliteration in Figure 1). Note that OCR errors are corrected directly into the modern orthography (e.g. 'ro' is transformed into 'v' instead of 'w').

```
<rules>
  <obligatory>
    <str find="à" replace="ā"/>
    <str find="ah" replace="ā"/>
    <str find="w" replace="v"/>
    <str find="tsch" replace="č"/>
    <str find="sch" replace="š"/>
    <str find="ees" replace="ies" match="end"/>
  </obligatory>
  <optional>
    <str find="ro" replace="v"/>
    <str find="a" replace="ā"/>
    <str find="l" sensitive="yes">
      <replace>I</replace>
      <replace>J</replace>
    </str>
  </optional>
</rules>
```

Figure 2: A set of sample transliteration rules.

[3] The system will provide back-end services for reading aids (in a form of pop-up menus) in the web-interfaces of the NLL digital collections.

[4] The longest substring not necessarily is the preferable one.

For any rule it is possible to add additional requirements that it is applied only if the target string matches the beginning or the end of a word, or an entire word, and/or that the rule is case-sensitive.

Transliteration rules are provided to the engine via an external configuration file. The current implementation of the engine allows providing several alternative rule sets. An appropriate set of rules can be chosen automatically, based on the document's metadata, e.g. typeface, publication year and type (a book or a newspaper). For the NLL corpus, currently two separate rule sets are being used: one tailored for texts in the Fraktur typeface printed after year 1880, and the other – for texts in the Latin typeface starting from the first item until the transition to the modern spelling in 1930-ies. A work in progress is to develop a set of rules for earlier Fraktur texts of 1750–1880. In future, the rule sets can be easily specialized if it will be experimentally verified that it would be advantageous to remove (or add) some transformation rules, for example, when processing documents of 1920-ies.

3.2 Applying transliteration rules

When the transliteration engine is started, each set of rules is loaded into the memory and is stored in a hash map using the target strings as keys. This gives us the ability to access all the possible replacements for a given target string in effectively constant time[5].

Transformations are performed with the help of dynamic programming and memorization. Each token is processed by moving the cursor character by character from the beginning to the end. In each position we check if characters to the left from the cursor correspond to some target string. In an additional data structure we keep all transformation variants for the first character, for the first two characters, for the first three characters etc. The transformation variants for the first *i* characters are formed as follows (consult Figure 3 for an example):

- For every rule whose target string matches the characters from the *k*-th position till the *i*-th position, a transformation variant (for the *i*-th step) is formed by concatenating each transformation variant from the *k*-th step with the rule's replacement string.
- From each transformation variant in length *i*-1 form a transformation variant in length *i* by adding the *i*-th character from the original token if there is no obligatory rule with a target string matching the last character(s) to the left from the cursor.

When the cursor reaches the end of the string, the obtained transformation variants are sorted in two categories: "more trusted" variants that are produced by the obligatory rules only, and "less trusted" variants that are produced also by the optional rules.

In Figure 3, it appears that "dāroanām" is a more trusted variant than "dāvanām", although actually it is vice versa. The false positive variant is eliminated in the next processing step, while the other one is kept (see Section 3.3).

Input: dahroanàm			
Step 1:	d	Step 5:	dāro, dāv
Step 2:	da, dā	Step 6:	dāroa, dāva, dāroā, dāvā
Step 3:	dā, dā	Step 7:	dāroan, dāvan, dāroān, dāvān
Step 4:	dār	Step 8:	dāroanā, dāvanā, dāroānā, dāvānā
Output (Step 9): dāroanām, dāvanām, dāroānām, dāvānām			

Figure 3: Sample application of transliteration rules. The input comes from Fig. 1 (line 2, token 6). Consult Fig. 2 for the rules applied (producing the underlined strings).

To speed up the transliteration, it is possible for user to instruct the engine not to use the optional rules for the current token.

3.3 Verifying transliteration variants

If transliteration is performed in the way it is described in the previous section, it produces plenty of nonsense alternatives. Thus we need a technique to estimate which of the provided results is more credible. One such estimate is implicitly given by the differentiation between obligatory and optional rules.

Another way to deal with this problem is to obtain a large list of known valid words and check the transliteration variants against it. Typically these would be lists of headwords from various dictionaries, however, due to the rich morphological complexity of Latvian, word lists, in general, are not very usable in a straightforward manner, but we can use a morphological analyzer to obtain the potential lemmas for the acquired transformation variants.

The exploited analyzer (Paikens, 2007) is based on a modern and rather modest lexicon (~60 000 lexemes) – although a lot of frequently used words are the same in both modern and historical Latvian, there is still a large portion of words out of vocabulary. Therefore we use a suffix-based guessing feature of the analyzer to extend its coverage when the lexicon-based analysis fails.

Transliteration variants whose lemmas are found in a list of known words are considered more credible. Currently we use wordlists from two large Latvian on-line dictionaries: one that primarily covers the modern lexicon (~190 000 words, including regional words and proper names), and one that covers the historical lexicon (>100 000 words, manually transliterated in the modern orthography). To extend the support for proper names (surnames and toponyms), we also use the Onomastica-Copernicus lexicon[6].

In the whole transliteration process we end up with six general credibility groups for the transliteration variants:

1. Only the obligatory rules have been applied; lemmatization has been done without guessing;

[5] This is important for the future use-cases where the service will provide probabilistic full-text transliteration.

[6] http://catalog.elra.info/product_info.php?products_id=437

the lemma is found in a dictionary.

2. Only the obligatory rules have been applied; lemmatization has been done by guessing; the lemma is found in a dictionary.
3. At least one optional rule has been applied; lemmatization has been done without guessing; the lemma is found in a dictionary.
4. At least one optional rule has been applied; lemmatization has been done by guessing; the lemma is found in a dictionary.
5. Only the obligatory rules have been applied; the lemma could not be verified by a dictionary.
6. At least one optional rule has been applied; the lemma could not be verified by a dictionary.

For instance, if we take the variants from Figure 3, “dāroanām” is not found in the morphological lexicon and by guessing it might be lemmatized as “dāroana” (noun) or “dāroant” (verb) – none of these nonsense words can be found in a dictionary. However, “dāvanām” is both recognized by the morphological lexicon as “dāvana” (‘gift’) and is found in a dictionary. A sample of full output data that is returned by the transliteration and lemmatization service is given in Figure 4.

```
<translit input="dahroanàm">
  <group opt_rules="no" guess="no" dict="yes"/>
  <group opt_rules="no" guess="yes" dict="yes"/>
  <group opt_rules="yes" guess="no" dict="yes">
    <variant wordform="dāvanām">
      <lemma form="dāvana">
        <dict id="MEV"/>
        <dict id="SV"/>
      </lemma>
    </variant>
  </group>
  <group opt_rules="yes" guess="yes" dict="yes">
    <variant wordform="dāvānām">
      <lemma form="dāvāna">
        <dict id="MEV"/>
      </lemma>
    </variant>
  </group>
  <group opt_rules="no" dict="no">
    <variant wordform="dāroanām"/>
  </group>
  <group opt_rules="yes" dict="no">
    <variant wordform="dāroānām"/>
  </group>
</translit>
```

Figure 4: Sample output data returned by the transliteration and lemmatization service.

Usually each of these groups contain more than one variant, thus it would be convenient to sort them in a more relevant order, e.g. by exploiting wordform frequency information from a text corpus. For instance, “dāvāna” (in Figure 4) is a specific orthographic form of “dāvana”; it is not used in modern Latvian and is rarely used even in historical texts.

First, a reasonable solution (at the front-end) would be that variants that are verified by a dictionary are given to the end-user before other variants – such approach is justified by our preliminary evaluation (see Section 4). The verified variants that are found in a large on-line dictionary (tagged by ‘SV’ in Figure 4) can be further passed to the dictionary service to get an explanation for the possible meanings of the word (see Section 5).

Second, a pragmatic trade-off would be that lemmas that are obtained by applying the optional rules and are not found in any dictionary are not included in the final output to avoid overloading end-users with too many irrelevant options (again, see Section 4).

3.4 Alternative sets of transliteration rules

Linguists distinguish several general groups in which Latvian historical texts can be arranged according to the orthography used.

In the current architecture, the transliteration service receives a single wordform per request along with two metadata parameters: publication year and typeface (Fraktur or Latin). Publication type (a book or a newspaper) could be added if necessary.

Taking into account the general groups and the provided metadata, for each case there should be a specific, handcrafted set of transliteration and OCR correction rules. The metadata theoretically could be used for automatic selection of a rule set. However, in practice it cannot be guaranteed (considering an isolated wordform) that the selection is the most appropriate one, if all the parameters overlap between two groups (due to the fact that several historical orthography variants were used in parallel for a prolonged time, and changes were rather gradual). There is also an objective issue caused by the uniform OCR configuration that has been used for all texts in the mass-digitalization despite the orthographic variations. In the result, all potential rule sets would have to extensively deal with OCR errors overgenerating transliteration variants in order to improve recall. Therefore we have defined only two general rule sets: one for the Fraktur script, and one for the early Latin script (see Section 3.1 for more detail).

Theoretically, there are at least two (parallel) scenarios how this issue could be addressed in future. First, a specific OCR configuration (a FineReader training file) could be adjusted for each text group, running the OCR process again and enclosing configuration IDs in the metadata. To a large extent, this could be done automatically, involving manual confirmation in the borderline cases. However, our experiments with FineReader 11 show that this would not give a significant improvement[7] and would not scale well over different facsimiles of the same group, i.e., it would not be cost-effective. Second, a larger text fragment could be passed along with the target wordform, so that it would

[7] For a book (1926) fragment, the accuracy in both cases is about 95% at the letter level and about 75% at the word level.

be possible to detect specific orthographic features by frequency analysis of letter-level n-grams and by analyzing the spelling of common function words. This would allow choosing an optimal set of transformation rules to ensure an optimal error correction and transliteration[8]. More tailored sets of rules should also decrease the amount of nonsense transliteration variants.

3.5 Disambiguation – a future task

The transliteration system, as described above, results in multiple options for possible modern spellings of a given wordform. While this is a usable approach in interactive use-cases for which the system has been initially designed, other applications that require full-text transliteration most likely require automatic disambiguation as well, receiving a single, most probable variant for each wordform.

A naive probability ranking could be obtained by comparing the variants against a word frequency table obtained from a modern text corpus of a matching genre (i.e., newspapers, fiction etc.), according to the metadata of the analyzed text. A more reasonable approach would be exploitation of a POS tagger of modern Latvian[9] to eliminate part-of-speech categories that are contextually unlikely possible. In addition, a word-level n-gram model of modern Latvian could be used, but there might be a lot of rarely used or out-of-vocabulary words, particularly in the case of the NLL newspaper corpus that includes a large number of proper names. The problem of transliteration can be also seen as a problem of machine translation between very similar languages. Statistical phrase-based techniques could be applied, similarly as it has been done for multilingual named entity transliteration (Finch & Sumita, 2009), however, it would require a parallel corpus.

4. Evaluation

The performance of each transformation rule set can be estimated by comparing an automatic transliteration of a historical text with a manually verified transliteration of the same text. We have identified several historical books that have been reprinted in the modern orthography with minor grammatical or lexical changes to the language. We have semi-automatically aligned several book chapters, and we have also manually transliterated several pages from newspapers of various time periods to obtain a small, but a rather representative tuning and test corpus (see Figure 5).

For the current target application – a reading aid for historical texts – we have evaluated the performance of the multi-option transliteration, attempting to minimize the number of variants that are returned while maximizing the accuracy rate – that the known correct variant is among the returned ones.

Year	Title	Type	Tokens
1861	*Latviešu avīzes*	newspaper, early Fraktur	1025
1888	*Lāčplēsis*	book, early Latin	4308*
			917
1913	*Mērnieku laiki*	book, Fraktur	2880*
			5438
1918	*Baltijas ziņas*	newspaper, Fraktur	1001

Figure 5: A parallel corpus used for tuning (*) and evaluation of transliteration rules.

The tuning corpus identified a number of additional historical spelling variations, and several systematic OCR mistakes that can be corrected with transliteration rules. Figure 6 shows the final performance on the tuning corpus. The results clearly show the importance of the dictionary-based verification and that it would not be reasonable to overload the end-users with the over-generating variants that are acquired by optional rules and that are not verified by a dictionary (`no_dict`, `opt_rules`). The other credibility groups give 97% accuracy on the tuning corpus with 2.77 variants per token.

Credibility group	Accuracy	Variants
`dict`, `no_opt_rules`, `no_guess`	55.6 %	0.63
`dict`, `no_opt_rules`, `guess`	6.1 %	0.14
`dict`, `opt_rules`, `no_guess`	31.1 %	0.73
`dict`, `opt_rules`, `guess`	3.7 %	1.00
`no_dict`, `no_opt_rules`	0.5 %	0.27
`no_dict`, `opt_rules`	1.4 %	30.61
No variant produced:	1.6 %	0

Figure 6: Evaluation on the tuning corpus: an average number of variants and accuracy (contains the correct variant) per credibility group (consult Section 3.3).

These results also indicate a ceiling for the possible accuracy of this method at around 98%, no matter how well the transliteration rules are improved. Manual review of unrecognised words shows that around 1% of words have been irreparably damaged by OCR, and around 1% of words are unique and out of vocabulary: foreign words, rare proper names etc., where many equally likely transliteration options would be possible.

Note that lemmatization by guessing has been necessary “only” in about 10% cases – the common word lexicons of historical and modern Latvian highly overlap.

In the evaluation we are counting only exact spelling matches (including diacritics), and we are counting only word tokens (excluding numbers, punctuation etc.). The evaluation of transliteration accuracy for various texts is shown in Figure 7.

Year	Type	Accuracy	Variants
1861	newspaper, Fraktur	87.7 %	3.12
1888	book, Latin	96.7 %	2.45
1913	book, Fraktur	96.6 %	3.19
1918	newspaper, Fraktur	88.8 %	2.81

Figure 7: Evaluation on the test corpus.

[8] This would even allow distinguishing more specific rule sets than it is possible by relying only on the (extended) metadata.

[9] e.g., http://valoda.ailab.lv/ws/tagger/ or the one developed by Pinnis and Goba (2011).

We have observed that the OCR mistakes in the NLL corpus can be tackled by the same means as orthography changes, significantly improving the output quality: from around 75% word-level accuracy in the source texts (books) to around 88% (for newspapers) and 97% (for books) after transliteration. The correlation between font-face changes and orthography developments, as well as the possibility to match the transformation results against a large lexicon allows tackling both problems simultaneously.

However, as the evaluation shows a significant accuracy difference between book and newspaper content, we have analyzed the structure of all identified errors. The errors have been grouped as unrepaired OCR mistakes, unrepaired lexical or spelling differences in the historical language, and errors in transliteration rules, as shown in Figure 8. This indicates that the technique is vulnerable to scanning quality (as the *Baltijas ziņas* facsimile is of a comparatively low quality), and that there is still a future work to be done in improving the lexical change repair rules for the 1860-ies and earlier texts.

Error type	***Latviešu avīzes***	***Baltijas ziņas***
OCR mistakes	28 (23.5%)	83 (74.1%)
Lexical differences	83 (69.8%)	12 (10.7%)
Malfunctioning rules	8 (6.7%)	13 (11.6%)
Other	0	4 (3.6%)
Total	119	112

Figure 8: Error analysis.

5. Dictionary service

On a user request, an unknown word (lemmatized in the modern orthography by the transliteration service) is passed to a dictionary service that is based on a large on-line dictionary of Latvian[10]. The dictionary contains nearly 200 000 entries that are compiled from the Dictionary of the Standard Latvian Language[11] and more than 180 other sources. It covers common-sense words, foreign words, regional and dialect words, and toponyms (contemporary and historical names of regions, towns and villages in Latvia). Explanations include synonyms, collocations, phraseologies and historical senses.

```
<dict id="SV">
  <entry src="KV">
    <word>
      <lemma>dāterēt</lemma>
      <gram>apv.</gram>
    </word>
    <sense>
      <def>Ātri un neskaidri runāt.</def>
    </sense>
  </entry>
</dict>
```

Figure 9: An entry returned by the dictionary service.

A simple entry returned by the dictionary service is given in Figure 9. It gives a meaning of a rarely used historical regional word for which even Google returns no hits (as of 2012-04-01).

6. Use-cases

The initial and primary goal is to integrate these services in the interactive user interface of an on-line digital library of historical periodicals[12], allowing users to get hints on what a selected utterance of a (historical) word means.

A future goal is to facilitate extraction and cataloguing of named entities in historical corpora. For this purpose, the transliteration engine will be integrated in a named entity recognition system that is currently being developed[13]. It will be used while indexing person names and other named entities mentioned in texts by mapping these names to their modern spelling. This will allow searching for proper names regardless of how they might be spelled in the historical documents.

7. Conclusion

We have designed and implemented a set of services that facilitate the accessibility of historical Latvian texts to contemporary readers. These services will be used to improve the accessibility of historical documents in the digital archives of the National Library of Latvia – a sizeable corpus containing about 4 million pages[14].

Our preliminary evaluation shows that the rule-based approach with dictionary verification works well even with a single rule set for all Fraktur texts, returning 2.89 variants in average with a possibility of 92.45% that the correct one is among them. Period-specific tuning of transliteration rules can raise the accuracy up to 96.5% for both books and newspapers.

A future task is to provide an automatic (statistical) context-sensitive disambiguation among these variants.

It has to be noted that the system is designed to be generic and extensible for other transliteration needs by specifying appropriate sets of lexical transformation rules. While currently it is aimed to be used for analysis of historical texts, future work could address the transliteration of modern texts in cases where different spelling is systematically used. For instance, transliteration to the standard language is necessary in the case of user-generated web content (comments, tweets etc.) where various transliteration approaches for non-ASCII characters have often been used in Latvian due to the technical incompatibilities and inconvenience of various systems or interfaces.

Acknowledgments

The preparation of this paper has been supported by the European Regional Development Fund under the project

[10] http://www.tezaurs.lv/sv/

[11] Latviešu literārās valodas vārdnīca. Vol. 1–8. Riga: Zinātne, 1972–1996 (>64 000 entries).

[12] http://www.periodika.lv/

[13] Unpublished work, expected to be ready by the end of 2012.

[14] It is expected that a working demo of these reading aids will be available in May 2012.

No. 2DP/2.1.1.2.0/10/APIA/VIAA/011. The authors would like to thank Artūrs Žogla from the National Library of Latvia and the anonymous reviewers for their comments.

References

Finch, A., Sumita, E. (2009). Transliteration by Bidirectional Statistical Machine Translation. In *Proceedings of the 2009 Named Entities Workshop (NEWS)*, Suntec, pp. 52–56

Ozols, A. (1965). *Veclatviešu rakstu valoda*. Riga: Liesma

Paegle, Dz. (2001). Latviešu valodas mācībgrāmatu paaudzes. Otrā paaudze 1907–1922. In *Teorija un prakse*. Riga: Zvaigzne ABC, pp. 39–47.

Paegle, Dz. (2008). Pareizrakstības jautājumu kārtošana Latvijas brīvvalsts pirmajos gados (1918–1922). In *Baltu filoloģija XVII, Acta Universitatis Latviensis*, pp. 89–102

Paikens, P. (2007). Lexicon-Based Morphological Analysis of Latvian Language. In *Proceedings of the 3rd Baltic Conference on Human Language Technologies (Baltic HLT 2007)*, Kaunas, pp. 235–240

Pinnis, M., Goba, K. (2011). Maximum Entropy Model for Disambiguation of Rich Morphological Tags. In *Proceedings of the 2nd Workshop on Systems and Frameworks for Computational Morphology*, Communications in Computer and Information Science, Vol. 100, Springer, pp. 14–22

Tanner, S., Muñoz, T., Ros, P.H. (2009). Measuring Mass Text Digitization Quality and Usefulness: Lessons Learned from Assessing the OCR Accuracy of the British Library's 19th Century Online Newspaper Archive. *D-Lib Magazine*, 15(7/8)

Zogla, A., Skilters, J. (2010). Digitalization of Historical Texts at the National Library of Latvia. In I. Skadiņa, A. Vasiļjevs (Eds.), *Human Language Technologies – The Baltic Perspective (Baltic HLT 2010)*, Frontiers in Artificial Intelligence and Applications, Vol. 219, IOS Press, pp. 177–184

Text encoding and search for Āyurvedic texts: An interconnected lexical database

Baldev Ram Khandoliyan, Rajneesh Kumar Pandey, Archana Tiwari, Girish Nath Jha

Special Centre for Sanskrit Studies,
Jawaharlal Nehru University, New Delhi-67
{baldevramjnu, rajneesh1988, archana.jnu, girishjha} @gmail.com

Abstract

The paper explores an interconnected lexical resource system for key texts of Ayurveda. The system which is in the form of an online index can be tested at http://sanskrit.jnu.ac.in/ayur/index.jsp. The paper also discusses the text encoding mechanisms and search processes that have been used to create the resource. Though *Āyurveda* has had a long tradition of texts and commentaries, we have taken only two key texts - Suśruta Samhitā and Caraka Samhitā and a glossary called Bhāvaprakāśa Nighaṇṭu with Amarakośā. The system works as an interactive and multi-dimensional knowledge based indexing system with search facility for these mainstream Ayurvedic texts and has potentials for use as a generic system for all āyurvedic texts which have similar structure.

Keywords: Ayurvedic lexicon, Ayurveda, Caraka Samhitā (CS), Suśruta Saṁhitā (SS), Bhāvaprakāśa Nighaṇṭu (BPN), Vanauṣadhivarga (VV), Amarakośā (AK), Āyurvedic herbs, indexing

1. Introduction

India has a rich heritage in traditional medicinal systems like Ayurveda which can be called as the most ancient documented on medical healing and healthcare. It is a holistic time tested Indian system of healing and is also a complete system for restoring, maintaining and enhancing health. It advocates use of natural healing methods tailored to the individual and advocates preventive measures through diet, lifestyle and medicines as customized therapy. On the whole, Āyurveda maintains the body in a balanced state of health using natural cures.

Āyurveda was preserved and saved by the people of India as a traditional "science of life". In the earlier days of its conception, the system of Āyurvedic medicine was orally transferred via the *gurukula*(ancient Indian educational system) system until a written script came into existence which led to the documentation of famous Āyurvedic text slike CS, SS, Mādhava Nidāna, BPN etc.

2. Uses of the Database search for Āyurveda

The Āyurveda domain is vast and spread across various ancient texts. In-depth analysis of domain knowledge with integrated multidimensional view and unbiased authentic understanding is required. The indexing system of Āyurvedic texts can be used in various NLP applications like building WordNet, āyurvedic dictionaries, Sanskrit-Indian Language Machine Translation System (MTS) etc. Unique words are a basic need of a WordNet and a dictionary. Automatic indexing and sorting is a good tool to extract unique words from a given text. Though, there is not a direct use of indexing system in machine translation but can be helpful in generating a context lexicon[1] This work, besides being an essential resource in NL system of Sanskrit, may also be useful for authentic and referential knowledge about Indian heritage. The system can also be very useful for the researches of medical sciences by providing the facts from the huge text which cannot be otherwise easily read and understood.

3. Previous Work

For the emerging R & D area of Sanskrit informatics, it is necessary to make indices available online. Unfortunately, the task of electronic indices for Indian heritage has not attracted required attention of computational linguists. Some efforts made in this area, directly or indirectly, are listed here:

The Maharishi University of Management has created an index of Vedic Literature. Here Ayurvedic texts are available in the form of pdf files[2].

Center for Development of Advanced Computing

[1] a lexicon with the entries including the uncommon words frequently occurring in the context of use of that particular word, which, later assigning the sense may be used as a lexicon for word sense disambiguation.

[2] http://is1.mum.edu/vedicreserve/

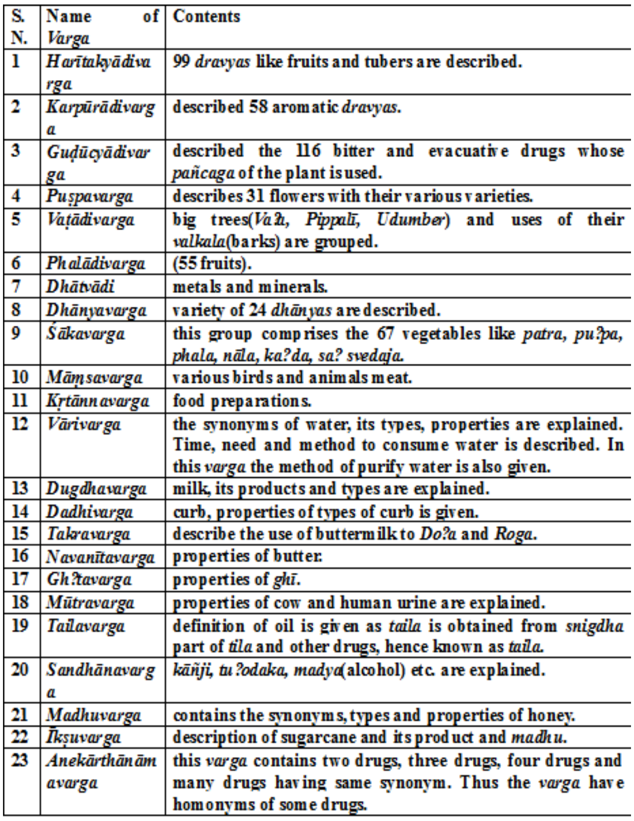
(CDAC), Pune(India) is developing a software for Āyurveda named AyuSoft. AyuSoft is a vision of converting classical Ayurvedic texts into comprehensive, authentic, intelligent and interactive knowledge repositories with complex analytical tools. Much of the details on it have not been provided at the link[3].

4. Distinction of Present System over Previous Ones

The online Āyurvedic texts at Maharishi University of Management are available only in the form of pdf files with proprietary fonts. This makes it difficult to use data for any meaningful search. We are planning to have discussions with them to obtain their text repository. AyuSoft, the CDAC initiative is under development. Our system is a dynamic search indexer which provides three kinds of search facility – string input search, search by listing of words by first letter and browsing the word by texts structure.

5. Important texts of Āyurveda

Present work has considered three traditions of Āyurveda in India — two of them are based on the compendia of CS and SS, and a third tradition known as Kāśyapas. Both the SS and CS are the products of several editorial hands, having been revised and supplemented over a period of several hundred years. Aṣṭāṅga Hṛdayam byVāgbhaṭa is a synthesis of earlier Ayurvedic materials. Another work associated with the same author, the Aṣṭāṅga Saṃgraha, contains much the same material in a more diffuse form, written in a mixture of prose and verse. The relationship between these two works, and a third intermediate compilation, is still a topic of active research. The works of Caraka, Suśruta, and Vāgbhaṭa are considered canonical and reverentially called the Vṛhad Trayī. In the early 8th century, Mādhava wrote his Nidāna, a work on etiology, which soon assumed a position of authority. He lists diseases along with their causes, symptoms, and complications. The precise description of *dravyas* or medicinal plants, right from the Vedic period was grouped under the Āyurvedic Nighaṇṭu texts. Nighaṇṭus are basically the specific lexical texts dedicated for the study of all aspects of drugs (herbs and plants) from their place of origin, their pharmacological actions, useful parts up to preparations and dosage. The Nighaṇṭu may be defined as a glossary containing synonymous groups, the names of drugs, plants, animals minerals and that is administered either as food or medicine to the human body. Such kind of lexicons are Dhanvantarinighaṇṭu, Prayāya-ratnamālā, Prayāyamuktāvalī, Nighaṇṭu Śeṣa, MadanVīnoda, KeyadevNighaṇṭu, Rāj Nighaṇṭu (Abhidhānacintāmaṇi or Nighaṇṭurāja), Bhāvaprakāśa Nighaṇṭu Śivakośa,śabda Candrikā, Dakṣināmūrti Nighaṇṭu, Dravyamuktāvalī, Prayāyārṇav etc. BPN is an important work of Āyurveda, which is enumerated among Laghutrayī. Though it is mentioned as the third text among Laghutrayī, it is a popular work among *Vaidyas* for Centuries. It is one of the classical works of Bhāvamiśra. The historians of Āyurveda consider Bhāvamiśra as a bridge between medieval period and modern period. In his work two portions are there - *Saṃhitā* portion divided into three parts like *Pūrvakhaṇḍa*, *Madhyamakhaṇḍa* and *Uttarakhaṇḍa*. Other one is the Nighaṇṭu portion, which is popularly known as Bhāvaprakāśa Nighaṇṭu. Bhāvamiśra has followed most of the Madanapāla Nighaṇṭu in this work. This Nighaṇṭu is considered as the latest among classical works in the field of Dravyaguṇa Nighaṇṭu. The Nighaṇṭu portion is commonly followed by physicians and students of *Dravyaguṇa*. This Nighaṇṭu consists a total of 23 chapters-



S. N.	Name of *Varga*	Contents
1	*Harītakyādivarga*	99 *dravyas* like fruits and tubers are described.
2	*Karpūrādivarga*	described 58 aromatic *dravyas*.
3	*Guḍūcyādivarga*	described the 116 bitter and evacuative drugs whose *pañcaga* of the plant is used.
4	*Puṣpavarga*	describes 31 flowers with their various varieties.
5	*Vaṭādivarga*	big trees(*Va?a, Pippalī, Udumber*) and uses of their *valkala*(barks) are grouped.
6	*Phalādivarga*	(55 fruits).
7	*Dhātvādi*	metals and minerals.
8	*Dhānyavarga*	variety of 24 *dhānyas* are described.
9	*Śākavarga*	this group comprises the 67 vegetables like *patra, pu?pa, phala, nāla, ka?da, sa? svedaja.*
10	*Māṃsavarga*	various birds and animals meat.
11	*Kṛtānnavarga*	food preparations.
12	*Vārivarga*	the synonyms of water, its types, properties are explained. Time, need and method to consume water is described. In this *varga* the method of purify water is also given.
13	*Dugdhavarga*	milk, its products and types are explained.
14	*Dadhivarga*	curb, properties of types of curb is given.
15	*Takravarga*	describe the use of buttermilk to *Do?a* and *Roga*.
16	*Navanītavarga*	properties of butter.
17	*Gh?tavarga*	properties of *ghī*.
18	*Mūtravarga*	properties of cow and human urine are explained.
19	*Tailavarga*	definition of oil is given as *taila* is obtained from *snigdha* part of *tila* and other drugs, hence known as *taila.*
20	*Sandhānavarga*	*kāñji, tu?odaka, madya*(alcohol) etc. are explained.
21	*Madhuvarga*	contains the synonyms,types and properties of honey.
22	*Īkṣuvarga*	description of sugarcane and its product and *madhu*.
23	*Anekārthānāmavarga*	this *varga* contains two drugs, three drugs, four drugs and many drugs having same synonym. Thus the *varga* have homonyms of some drugs.

Table1: Content of BPN

[3] http://pune.cdac.in/html/hbcg/ayusoft/ayusoft.asp

6. Methodologies for Preserving Texts

On one side, ayurveda has a great tradition of indexing and lexicography, but on other side, there is very little searchable online content to make such a huge text easily accessible. This search system will help the Āyurvedic scholars to resolve their problems like availability of texts, search facility with dictionary support. In the CS, SS and BPN indexing, comparative, analytical, descriptive and technological methodologies are used. The presently available versions of CS, SS and BPN evolved through multiple transformation and redaction by many scholars from time to time. As SS is very important text on surgery, there are many versions available at the time. For the development of the indexing system, the selection of the authoritative edition is always a challenge. We have taken the most authoritative editions of these texts and have converted them into electronic forms with the help of the the project staff of Dr Girish Nath Jha (one of the co-authors).

CS and SS has multi- tiered division in it. CS is divided in *sthāna*. Ever *sthāna* has chapters. *CS* has written in prose and poetry style. Therefore, each chapter has verses and *sūtras*. The hierarchy of CS is-

sthāna → *adyāya* →*sūtra*

SS is primarily divided into two *tantras*- *pūrva* and *uttara* and contains further division into *sthānas*. *Sthānas* finally consists of *adhyāyas* which are composed in sutras. *Sūtras* are in both form- prose and verse. Thus the hierarchy of the sūtras of the text is given below-

tantra → *sthāna* → *adhyāya* → *sūtra*

Converting these texts to a generic database format was the next big challenge. Let us take the example of CS and SS. The original text has been stored in separate database tables. Other information related with the text are stored in other tables in an inter-related architecture to provide complete reference for the searched query. The database has four tables. In the first table, the name of the *sthānas* are given. Second table has information about *adhyāya* number with *sthāna* number. In the third table, the name of *adhyāya* is given and in the fourth table, the *sutra* of the text with the id is stored.

Adhyāya_id	Sūtra_id	Sūtra_saṃhitā	Sūtra_pada	Adhyāya_id_sequencial

Table 2: Main table named 'sūtra' of database

Sthāna_id	Adhyāya_id	Adhyāya_name

Table 3: Structure of adhyāya table

Tantra_id	Tantra_name	Sthāna_id	Sthāna_name

Table 4: Structure of sthāna table

7. Development of Ayurvedic Indexer

A dynamic search engine cum-indexer has been developed which is built in the front-end of Apache Tomcat Web server using JSP and Java servlets. It has its data in Unicode data files along with RDBMS in MS SQL server. For connecting the front-end to the database server the MS-JDBC connectivity has been used. The architecture of the system is as follows:

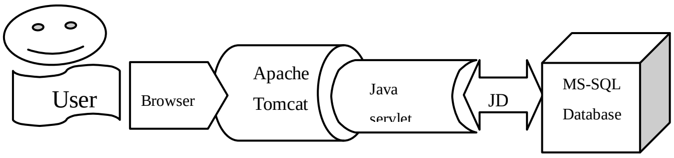


Fig. 1: Multi-tiered architecture of indexing system

7.1. Process Flow of the System

There are three ways to give input to the system e.g. Direct Search, Alphabet search and Search by the structure of the text in Devanāgarī UTF-8 format.

Step I: Preprocessing. Preprocessing a word mainly consists of transformation of a raw data required to facilitate further processing. For example – processer will remove any non Devanāgarī characters, punctuations that may have been inadvertently introduced by the user like "#" in CS and other similar cases.

Step II: Āyurveda Search and Database. At this step, the indexer makes an indexed list of exact and partially matching words. Getting the query as an input, the indexer, after a light preprocessing, sends it to the database. If the word has its occurrence in the database, the system gives the output.

Step III: Output level-1. At this stage, the indexer gives all the occurrences of the searched query with its numerical reference in a hyperlinked mode.

Step IV: Output level-2. Clicking on hyperlinked word, system shows its original place in the *śloka*(verse) [4]and also gives its full reference in the text. It also asks for further information from other online lexical resources.

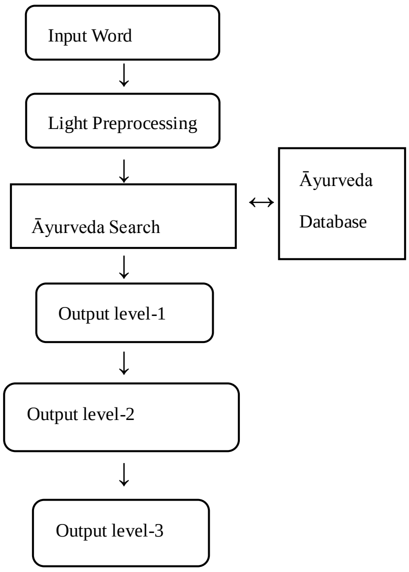


Fig.2:Process flow of the system

Step V: Output- final level. Here, the indexer gives a list of online lexical resources like Amarakosha(JNU) [5], "Spoken Sanskrit" Dictionary (by Klaus Glashoff, Germany)[6], Chicago university Apte dictionary, Chicago university Macdonell dictionary etc. and also gives the facility to do morphological analysis of the query with the help of POS tagger[7] and *subanta*(nominal inflectional morphology) analyzer[8].

7.2. Front-End of the Ayurveda Search

The front–end of the system is developed in UTF-8 enabled Java Server Pages (JSP) and HTML. The front-end of the software enables the user to interact with the indexing system with the help of Apache Tomcat web-server. The JSP technology helps to create web based applications combining Java code and displays the results as HTML. The snapshots of the indexing system are as follow:

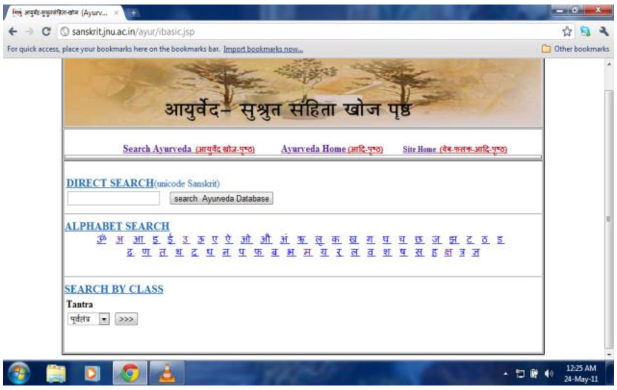

sanskrit.jnu.ac.in/ayur/ibasic.jsp
आयुर्वेद– सुश्रुत संहिता खोज पृष्ठ
Search Ayurveda (आयुर्वेद खोज-पृष्ठ)
Ayurveda Home (आदि-पृष्ठ)
Site Home (वेब-फलक-आदि-पृष्ठ)
DIRECT SEARCH(unicode Sanskrit)
search Ayurveda Database
ALPHABET SEARCH
SEARCH BY CLASS
Tantra
पूर्वतंत्र
>>>
12:25 AM
24-May-11

Fig.3: Search Page of the SS Indexing System

The indexer of the CS follows the same architecture. The screenshots of this indexer are given below –

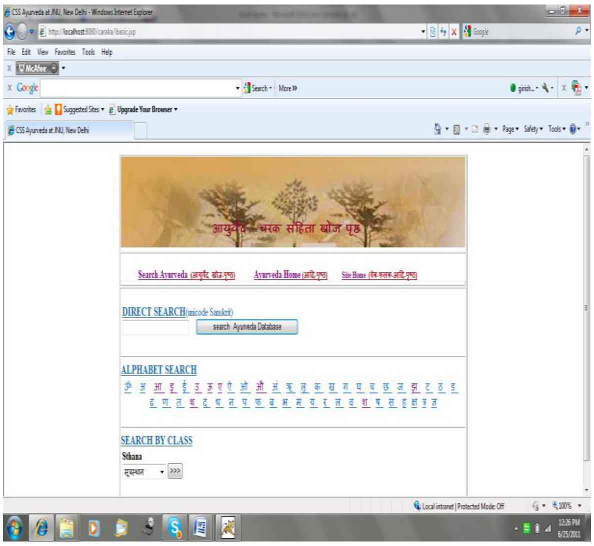
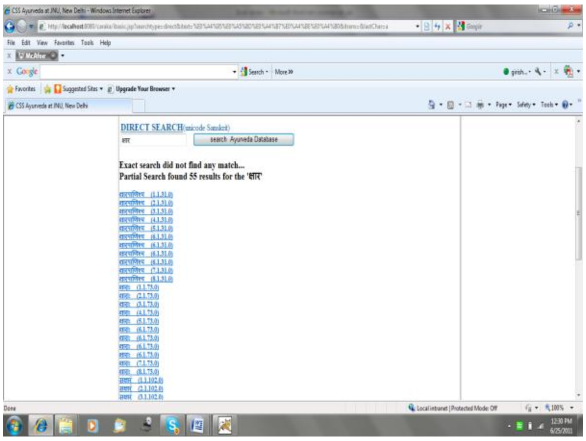

CSS Ayurveda at JNU, New Delhi - Windows Internet Explorer
http://localhost:8080/caraka/ibasic.jsp
आयुर्वेद – चरक संहिता खोज पृष्ठ
Search Ayurveda (आयुर्वेद खोज-पृष्ठ)
Ayurveda Home (आदि-पृष्ठ)
Site Home (वेब-फलक-आदि-पृष्ठ)
DIRECT SEARCH(unicode Sanskrit)
search Ayurveda Database
ALPHABET SEARCH
SEARCH BY CLASS
Sthana
सूत्रस्थान
>>>

Fig.4: Search Page of the CS Indexing System


CSS Ayurveda at JNU, New Delhi - Windows Internet Explorer
DIRECT SEARCH(unicode Sanskrit)
search Ayurveda Database

क्षार

Exact search did not find any match...
Partial Search found 55 results for the 'क्षार'

Fig.5: Second page as the result of searched input word in hyperlinked mode with their numerical references.

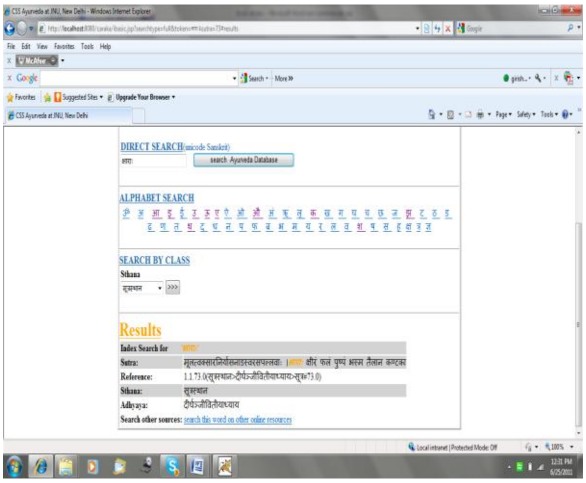
[4] *Śloka* is a term derived from Sanskrit. A *śloka* is a category of verse line developed from the Vedic *Anuṣṭubh*. *Śloka* s are usually composed in a specific meter. *Śloka* is a verse of two lines, each of sixteen syllables.

[5] http://sanskrit.jnu.ac.in/amara/viewdata.jsp

[6] http://www.spokensanskrit.de/index.php

[7] http://sanskrit.jnu.ac.in/post/post.jsp

[8] http://sanskrit.jnu.ac.in/subanta/rsubanta.jsp



DIRECT SEARCH(unicode Sanskrit)

अक्षः [search Ayurveda Database]

ALPHABET SEARCH

ॐ अ आ इ ई उ ऊ ए ऐ ओ औ अं ऋ लृ क ख ग घ च छ ज झ ट ठ ड ढ ण त थ द ध न प फ ब भ म य र ल व श ष स ह क्ष त्र ज्ञ

SEARCH BY CLASS

Sthana

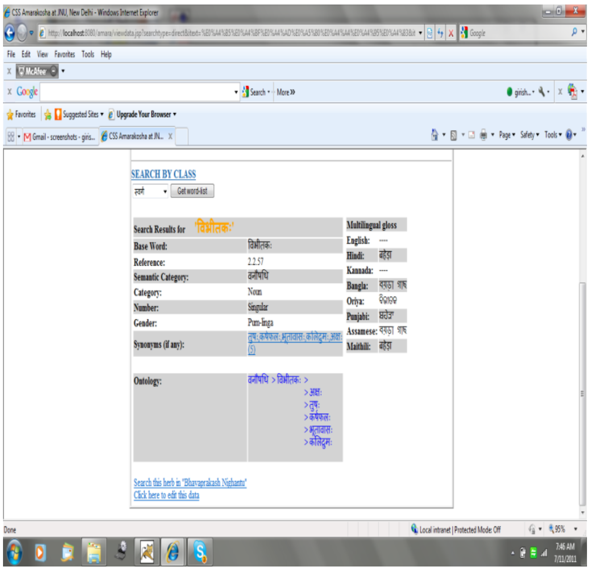
सूत्रस्थान [>>>]

Results

Index Search for	अक्षः
Sutra:	मूलत्वक्सारनिर्यासनाडस्वरसपल्लवाः । अक्षः क्षीरं फलं पुष्पं भस्म तैलान कण्टका
Reference:	1.1.73.0(सूत्रस्थान>दीर्घञ्जीवितीयाध्याय>सूत्र=73.0)
Sthana:	सूत्रस्थान
Adhyaya:	दीर्घञ्जीवितीयाध्याय
Search other sources:	search this word on other online resources

Fig.6: Final output of searched query with reference.

The indexer of the BPN is connected to the AK Search engine which can be accessed at Special Center for Sanskrit Studies, JNU, New Delhi[9] It takes input in three ways i.e. typing the input word, selecting input word through alphabetical category or selecting the input word from drop down box. The search starts in the Amarakośa database and if the search is an *auṣadhi* (herb) it will display as follows :



SEARCH BY CLASS

स्वर्ग [Get word-list]

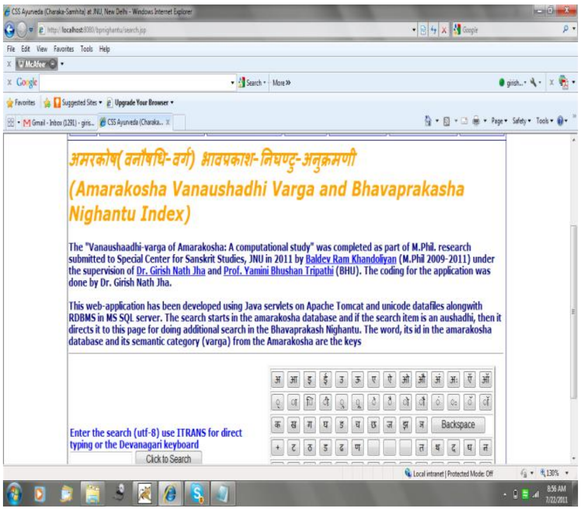
Search Results for	'विभीतकः'
Base Word:	विभीतकः
Reference:	2.2.57
Semantic Category:	वनौषधि
Category:	Noun
Number:	Singular
Gender:	Pum-linga
Synonyms (if any):	तुषः,कर्षफलः,भूतावासः,कलिद्रुमः,अक्षः (5)
Ontology:	वनौषधि > विभीतकः > > अक्षः > तुषः > कर्षफलः > भूतावासः > कलिद्रुमः

Multilingual gloss	
English:	----
Hindi:	बहेड़ा
Kannada:	----
Bangla:	বহেড়া গাছ
Oriya:	ବିଭୀତକ
Punjabi:	ਬਹੇੜਾ
Assamese:	বহেড়া গাছ
Maithili:	बहेड़ा

Search this herb in "Bhavaprakash Nighantu"
Click here to edit this data

Fig.7: First page as the result of *auṣadhi* searched in VV of AK

If the input word is related to VV then the page has a link for additional search as "*search this herb in Bhavaprakasha Nighantu*", On clicking this link, the system directs it to the BPN indexer as illustrated below-



अमरकोष(वनौषधि-वर्ग) भावप्रकाश-निघण्टु-अनुक्रमणी (Amarakosha Vanaushadhi Varga and Bhavaprakasha Nighantu Index)

The "Vanaushaadhi-varga of Amarakosha: A computational study" was completed as part of M.Phil. research submitted to Special Center for Sanskrit Studies, JNU in 2011 by Baldev Ram Khandoliyan (M.Phil 2009-2011) under the supervision of Dr. Girish Nath Jha and Prof. Yamini Bhushan Tripathi (BHU). The coding for the application was done by Dr. Girish Nath Jha.

This web-application has been developed using Java servlets on Apache Tomcat and unicode datafiles alongwith RDBMS in MS SQL server. The search starts in the amarakosha database and if the search item is an aushadhi, then it directs it to this page for doing additional search in the Bhavaprakash Nighantu. The word, its id in the amarakosha database and its semantic category (varga) from the Amarakosha are the keys

Enter the search (utf-8) use ITRANS for direct typing or the Devanagari keyboard

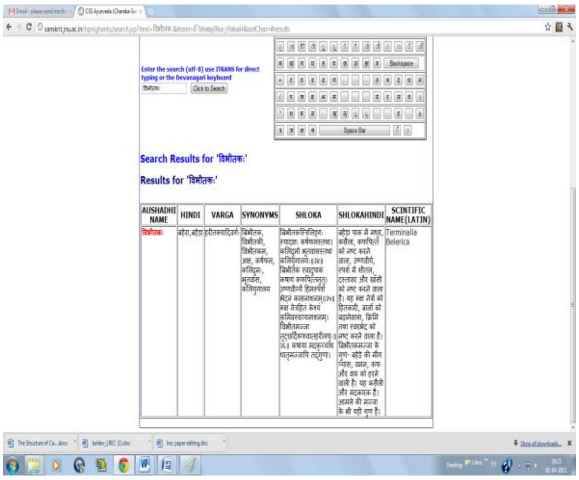
[Click to Search]

Fig.8: AK VV and BPN Indexer page

The input for this page is forwarded from the previous page. It also has the facility to accept a new input word through typing or in-built keyboard.

The database of the main page (AK Search engine) has an id assigned to each word of VV, this id of the input word is matched with BPN database to get additional input.

The system gives output on the basis of AK database, and BPN database. The output includes AUSHADHI NAME, HINDI NAME, VARGA, SYNONYMS, SHLOKA, SHLOKA HINDI, SCINTIFIC NAME(LATIN).



Enter the search (utf-8) use ITRANS for direct typing or the Devanagari keyboard

विभीतकः [Click to Search]

Search Results for 'विभीतकः'

Results for 'विभीतकः'

AUSHADHI NAME	HINDI	VARGA	SYNONYMS	SHLOKA	SHLOKAHINDI	SCINTIFIC NAME(LATIN)
विभीतकः	बहेरा,बहेड़ा	हरीतक्यादिवर्गः	विभीतक, विभीतकी, विभीतकम्, अक्ष, कर्षफल, कलिद्रुमः, भूतवास, कलियुगालय	विभीतकस्त्रिलिङ्गः स्यादक्षः कर्षफलस्तथा। कलिद्रुमो भूतवासस्तथा कलियुगालयः॥३४॥ विभीतकं स्वादुपाकं कषायं कफपित्तनुत्। उष्णवीर्यं हिमस्पर्शं भेदनं कासनाशनम्॥३५॥ रूक्षं नेत्रहितं केश्यं कृमिवैस्वर्यनाशनम्। विभीतमज्जा तृट्छर्दिकफवातहरो लघुः॥३६॥ कषायो मदकृच्चाथ धात्रीमज्जापि तद्गुणः।	बहेड़ा पाक में मधुर, कसैला, कफपित्त को नष्ट करने वाला, उष्णवीर्य, स्पर्श में शीतल, दस्तावर और खाँसी को नष्ट करने वाला है। यह रूक्ष नेत्रों को हितकारी, बालों को बढ़ानेवाला, क्रिमि तथा स्वरभेद को नष्ट करने वाला है। विभीतकमज्जा के गुण- बहेड़े की मींग प्यास, वमन, कफ और वात को हरने वाली है। यह कसैली और मदकारक है। आमले की मज्जा के भी यही गुण हैं।	Terminalia Belerica

Fig.9: Final output of the system

7.3 The Back-End of the System

The back-end of the indexing system consists of RDBMS, which contains correlative data tables. This Tomcat server based program connects to MS-SQL Server 2005 RDBMS through JDBC connectivity. These lexical resources are stored as Devanāgarī utf-8.

[9] http://sanskrit.jnu.ac.in/amara

There are three tables in database namely; “sūtra”, “adhyāya_name”, “adhyāya_no”, and “sthāna”. The descriptions of the tables have been discussed in the previous chapter. A design of the indexing of SS is given below-

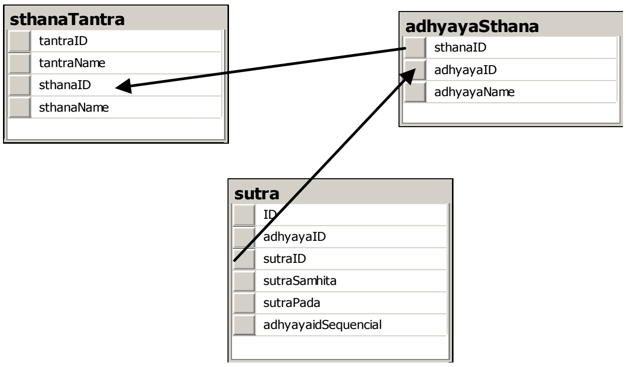
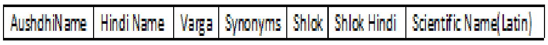


Fig10: System Module

The back end of server version also contains lexical resources in the form of text files. The data file is named lexicon.txt. It contains herbs of VV of AK and BPN in the following format:



AushdhiName	Hindi Name	Varga	Synonyms	Shlok	Shlok Hindi	Scientific Name(Latin)

Table.5: The format of data file

The back-end of the indexing system also consists of an RDBMS, which contains inter-related data tables. The lexical resources are stored as Devanāgarī utf-8 and data files. There are two tables namely; *Amarkosha*, *Bhavaprakasha*. The descriptions of the tables are as follows -

The basic database is *Amarkosha* which contains *tbl_basic* as the fundamental table connecting to the tables like *tbl_Category* and *tbl_Synonyms* for obtaining related information. In the *tbl_basic*, the column *id* is for the sequential id of the words, reference stores the textual reference from AK, *category* stores the id of the semantic category of the word (linked to the category table), *number*, *gender* store these information respectively. The remaining 8 columns store the multilingual glosses for the AK word.

The table *Bhavaprakasha* has a description of the herbs in *AushadhiName, Hindi Name, Varga, Synonyms, Shloka, Shloka Hindi, Scientific Name(Latin),* columns. The searches starts by the name (partial or full) of the respective herb from the AK database and searches it in the BPN database with column *AushadhiName*. The structure for database storage in the table is as follows:

AUSHADHINAME	HINDI NAME	VARGA NAME	SYNONYMS	SHLOKA	SHLOKA HINDI	SCINTIFIC NAME(LATIN)
विभीतकः	बहेरा, बहेड़ा	हरीतक्यादिवर्गः	बिभीतक, विभीतकी, विभीतकम, अक्ष, कर्षफल, कलिद्रुम, भूतवास, कलियुगालय	बिभीतकस्त्रिलिङ्गः स्यादक्षः ३४॥ बिभीतकं स्वादुपाकं कषायं कफपित्तनुत्। उष्णवीर्य्यं हिमस्पर्शं भेदनं कासनाशनम्॥३५॥	बहेड़ा पाक में मधुर, कसैला, कफपित्त को नष्ट करने वाला,… ..खाँसी को नष्ट करने वाला है।	Terminalia Belerica

Table .6: Structure of database storage, it is second table among two called ‘Bhavaprakasha’

7.4 Database Connectivity

The database connectivity has been done through the JDBC driver software. MS-JDBC Application Programming Interface (API) is the industry standard for database independent connectivity for Java with MS database. Since SQL server 2005 and JDBC support input and output in Unicode, this system accepts Unicode Devanāgarī text as well and prints result in the same format.[10]

8 Limitations of the System

- The system has fixed input and output mechanism. One can search his query in Unicode Devanāgarī only and the output will be in the same format. The work of transcoding of input and output in another format is being developed.
- The system can give result of those words, which are in split form in the database. For sandhi words, it tries a substring search to give all possible results.
- At present, the system is unable to give the translation in other languages.
- The system enables to search only string based query. It may fail to search synonymous words. It can give only index of those words which are exactly matched with the query. Future update will include linking it with Amarakośa (which has already been developed).
- The system does not provide the names of *dravyas*[11] in other Indian languages.

[10] http://java.sun.com/javase/technologies/database/
[11] Here *dravya* means ayurvedic herbs.

- The system is accessible on web only. It cannot be installed on user machines or delivered in a CD.

9 Conclusion

The paper discusses the dynamic indexing of some key Aurvedic texts. Such a facility on the web will encourage to the scholars of Sanskrit for taking interest in the area of research in *āyurveda*. The system is expected to help preservation of Sanskrit heritage texts. User can search the database by three different ways as mentioned above. One can search via providing direct input on the system or one can click on alphabets given on the system to search the specific keyword in the large pre-stored text. After clicking on the chapter link, it provides the list of indexed words in hyperlinked form. One can get full information about the indexed word by clicking on the indexed word. The system runs on the apache tomcat web server uses MS SQL server 2005 for database. The search engine is available on http://sanskrit.jnu.ac.in/ayur/index.jsp.

References:

1. Atrideva, (2007). *Suśruta Saṃhitā*, Motilal Banarasidas, Delhi

2. Aufderheide, A. C. Rodriguez-Martin, C. & Langsjoen, (1998). The Cambridge *Encyclopedia of Human Paleopathology*, Cambridge University Press,

3. Bakhariya, Aneesha, (2001). *Java Server Pages*, Prantice hall of India Pvt. Ltd., New Delhi,

4. Bhatia, Maj. Gen. S.L. (1972). *Medical sciences in ancient India*.
5. Briggs, Ricks, (1985) *Sanskrit and Artificial Intelligence* – NASA, Knowledge representation in Sanskrit and artificial Intelligence, California,

6. Chopra A and Doiphose W, (2002). *Ayurvedic medicine*: *Core concept, therapeutic principles, and current relevance*. Med. Clin. North Am.

7. Colebrooke, H.T. (1808). *Kosha or Dictionary of the sungskrita language by Umura singha with an English interpretations and annotations,* Serampore.

8. Date. C.J. (1987). *Introduction to Database System*, Addision Wesley.

9. Dharampal, (2000). *Indian Science and Technology in the Eighteenth Century*. Goa: Other India Press.
10. Dwivedi, Girish & Dwivedi, Shridhar, (2007). *History of Medicine: Suśruta- the Clinician - Teacher par Excellence*. National Informatics Centre (Government of India).

11. Dwivedi, Vishvanath, (2007). *Bhāvaprakāśa Nighaṇṭu* with hindi commentary, Motilal Banarashidas, delhi.

12. Gorilla, Vacaspati, (1997). *Saṃskrita Sāhitya kā Itihāsa*, Chowkhamba Vidya Bhavan, Varanasi.

13. Jha, Girish Nath (with R. Chandrashekar, Umesh Kumar Singh, Vibhuti Nath Jha, Satyendra Pandey, Surjit Kumar Singh, Mukesh Mishra), (Feb1, 2010). *Online Multilingual Amarakosha: the relational lexical database* demo paper in the proceedings of the 5th Global Wordnet Conference, IIT Mumbai

14. Jha, Girish Nath, (June1-3, 2005). *Information technology applications for Sanskrit lexicography: case of Amarakosha*, procs of the AsiaLex conference at National University of Singapore, Singapore

15. Katre Sumitra mangesh, (1937). *Indo Aryan Lexicography;* Poona Orientalist Vol.15.
16. Lamp, John and Milton, Simon, (2006). *Indexing research: An approach of information and knowledge resources, Information systems foundations : theory, representation and reality.*

17. Mettler, Cecilia. C. (1947). *A History of Medicine*, The Blackstone Company, Philadelphia, Toronto.
18. Mishra , Mukesh Kumar, (2012). *Computational Semantics for Sanskrit: The Case of Amarakosha Homonyms*, Lambert Academic Publisher, Germany

19. Pandeya, Gangasahay, (2010). *Bhāvaprakāśa Nighaṇṭu with hindi commentary*, Chowkhamba Bharati Akadami, Varanasi.

20. Sharma, Priya Vrat, (2004). *Suśruta Saṃhitā, Vol I* (*Sūtra sthāna*), Chaukhamba Visvabharati, Varanasi.

21. Sharma, Pt. Anantram, (2008). *Suśruta Saṃhitā*, Chaukhamba Surbharti Prakashan, Varanasi.
22. Singh, Amritpal, (2007). *Bhāvaprakāśa Nighauṇṭu with English translation*, Chaukhambha Orientalia, Delhi.

23. Singhal, Prof. G.D. and Colleagues, *Suśruta Saṃhitā*, Chaukhamba Sankrita Pratishthan, 2007(2nd edition)
24. Upadhyaya Baldev, (2001). *Saṃskrita Śāśtron kā Itihāsa,* Sharda Niketan, Varanasi.

25. Vakil, R.J. (1961). Romance of Healing and other Essays - *Our Medical Heritage*, Asia Publishing House, Bombay, calcutta, New Delhi, Madras, London,New York.

A Multitouch Enabled Annotation Editor for Digitized Historical Documents

Tobias Sippel, Jan-Torsten Milde

Hochschule Fulda, Computer Science Department
tobias.sippel@gmail.com, milde@hs-fulda.de

Abstract

In this paper we describe a system allowing to annotate digitized historical documents stored in METS/MODS format. The annoation is spacially connected to the original document. Both grafical and textual annoation are possible. The system is equipped with an intuitive touch based control and is designed to be a simple to use workbench for scientists working with historical texts.

Keywords: annotation editor, touch based control, digitized historical texts

1. Introduction

Over the last decade multiple tools for the creation of linguistic resources have been realized (see Kipp, 2001, Schmidt, 2004, Sasaki, 2002, Broeder et.al., 2001, Bird & Liberman, 1999). These tools have been developed to better support linguists within their daily work. Es- pecially the creation of multi level annotated corpora has been a cen- tral focus here. The tool development has also been strongly driven by the advancements in technology. As a central (text) technology the extensible markup language (XML) and its supporting process- ing standards (e.g. XPATH, XQuery, XSLT) have become a basis for most of the modern linguistic tools. In principle this should have lead to a higher interoperability between different tools. Unfortunately this is not the case as most tools are designed for a specific (linguistic) problem and therefore can not be easily used in conceptually differing research areas.

In our previous work we have been focusing on tools for the creation of phonetic corpora and multimodal corpora (TASX-Annotator, later the Eclipse-Annotator, see Milde & Gut, 2002, Behrens & Milde, 2006), followed by a tool for the creation of parallel text corpora (SAM, Geilfuss & Milde, 2006). The TASX-Annotator was used to create the LeaP Corpus (see Milde & Gut, 2002), collecting prosodic information of language learners. SAM has been developed to better support the creation of critical synopsises. In this case multi level commented versions of Goethe's Werther and Fontane's Effie Briest (see Seiler & Milde, 2003, Seiler & Milde, 2004) have been created.

From this work we have learnend, that tool creation is not successful without a tight cooperation with the linguistic community. At the same time this poses a problem, as the tools are only used by a very small number of people, a problem, that even gets bigger when it comes to research on cultural heritage objects.

2. The annotation editor

Based on this experience we have started to develop an annotation editor for digitized version of manuscripts and historical prints (see Sippel, 2011). The development is part of a cooperation between the Hochschul- and Landesbibliothek Fulda (HLB[1]) and the computer science department of Fulda University of Applied Sciences.

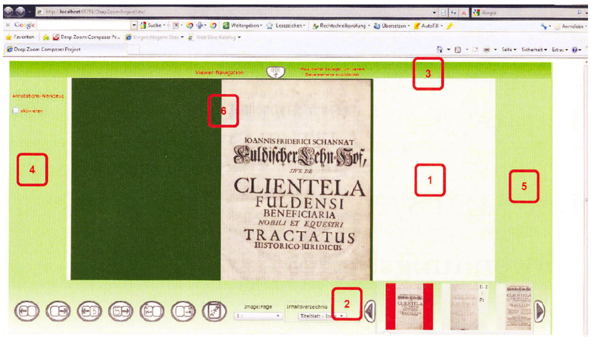


Figure 1: The user interface of the annotation editor

Currently the HLB is digitizing a large number of its historical documents. All documents are stored as high resolution, high quality uncompressed image files.

The librarians manually add bibliographical metadata, which allows to put the files into the central OPAC search catalog. The bibliographical data is stored in the XML-based MODS format (Metadata Object Description Schema), which in turn is part of a XML container file in METS format (Metadata Encoding and Transmission Standard, see MODS, 2011, METS, 2011). The METS/MODS files eventually carry all information about a specific digital artifact, including the logical structure of a text, the chaper information, and structural data with references to the actual digital images.

2.1. Storing annotations in METS/MODS

It is important to understand, that METS is an open container format. As such it is possible to add arbitrarily XML structured information into a METS file without corrupting the underlying format. The clear separation between the different parts of a METS file is realized by using XML namespaces.

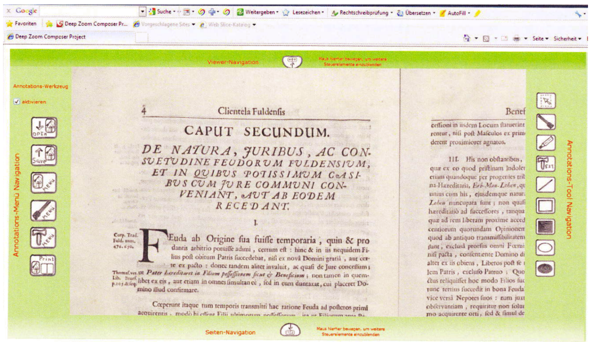
A METS document consists of seven parts. The METS header contains information about the document itself. Metadata about the creator or editor is found here. More Metadata is found in the next section (*descriptive metadata*), which could be stored internally and externally. The format allows to reference multiple files, allowing tools to add their specific metadata to a METS-encoded file. The following *administrative metadata* section stores informa-

[1] http://www.hs-fulda.de/index.php?id=169

tions like references to intellectual property data for the original files.

The next two sections are used by the annotation editor. In the *file section* all references to external files are stored. This section is also aiming at storing alternative versions of the current document. This is achieved by defining file groups, making it possible to add file groups that will only be interpreted by specific software tools and will be ignored by other tools.

The *structural map* section defines the hierarchical structure of the digital document. Here the different parts are linked together to form the final digital document. Again multiple versions can be defined here, which can be used to transparently store tool specific data in the file format. The final to sections (*structural links* and *behavior*) are currently not used by our system.



4 Clientela Fuldenſis

CAPUT SECUNDUM.

DE NATURA, JURIBUS, AC CONSVETUDINE FEUDORUM FULDENSIUM, ET IN QUIBUS POTISSIMUM CASIBUS CUM JURE COMMUNI CONVENIANT, AUT AB EODEM RECEDANT.

I.

Corp. Trad. Fuld. num. 476. 650.

Feuda ab Origine ſua fuiſſe temporaria, quin & pro dantis arbitrio potuiſſe admi, certum eſt: hinc & in iis nequidem Filius poſt obitum Patris ſuccedebat, niſi ex novâ Domini gratiâ, aut certe ex pacto: donec tandem aliter invaluit, ac quaſi de Jure conceſſum; ut *Pater hæreditaret in Filium poſſeſſionem ſicut & Beneficium*; non tamen in quemlibet ex eis, aut etiam in omnes ſimultaneè, ſed in eum duntaxat, cui placeret Domino illud confirmare.

Thomaſ. ver. Lib. Benef. p. 105. & ſeq.

Cœperunt itaque tum temporis transmitti hac ratione Feuda ad poſteros primi acquirentis,

Benef

ceſſioni in iisdem Locum ſtatuerint rentur, niſi poſt Maſculos ex prim derent proximiores agnatos.

III. His non obſtantibus, quæ ex eo quod priſtinam Indoler etiam quandoque per progenies trib na-Hæreditaria, *Erb-Man-Leben*, qu unius cum his, ejusdemque natur *Leben* nuncupata ſunt; non quaſi hæreditariò ad ſucceſſores, tanqua quæ ad rem liberam proxime acced centiorum quorundam Opinionem quod ab antiquo transmiſſibilitatem ſunt, excluſâ prorſus omni Fœmi niſi pacta, conſentiente Domino di alter ex iis obiens, Liberos poſt ſe Jem Patris, excluſo Patruo; Quo ctus reliquiſſet hoc modo Filios ſuc tunc tertius ſuccedit in bona Feuda vice versâ Nepotes ſuos: nam juxt obſervantiam, requiritur non ſolu mo acquirente orti, ſed & ſimul de

Figure 2: Zoomed in view of a double page

In the case of our tool the graphical and textual annoations of the (human) editor is added to the METS file. As this information is put into its specific namespace it becomes ”invisible” to other tools in a sense, that it does not interfere with their functionality. We chose to use XAML as the XML-format for storing the graphical annotations. XAML is a format for defining graphical user interfaces. As such it contains elements to precisely position text and graphics on a transparent canvas, which is put on top of the digitized image of the historical text. The following example shows the simplified XAML code for storing an annotation in a blue box at the position (223,42) of the current canvas.

```
<Canvas>
 <Rectangle
   Width="100" Height="100"
   Canvas.Left="223" Canvas.Top="42">
    <Rectangle.Fill>
      <SolidColorBrush Color="Blue"/>
    </Rectangle.Fill>
    <TextBlock>Annotation</TextBlock>
 </Rectangle>
</Canvas>
```

Figure 3: XAML code: annotation in a blue box

The efficient rendering of a canvas is part of the implementation technology (Silverlight and C#), which makes the editing functions of the tool fast and robust.

As the XAML based annotations and the METS based document structure both refer to the original images files it becomes relatively simple to transform the annotation into a seperate XML file which preserves the spacial relations.

2.2. Structured annotations and multilevel annotations

Currently the annotation editor only allows to *visually* annotate the document. In other words graphical and textual markers can be put on top of each page at arbitrary positions. No restrictions or rules are provided by the system that guide the annotator. We found this approach to be the most natural and versatile for most scientists. The digitized document can be treated just like high quality digital photo copy of the original document.

As a drawback, this approch could lead to a highly personalized and therefore incomparable, isocratic annotation style, which could create problems, when it comes to a computer supported analysis of the annotation data. A more structural approach to annotating the documents would therefore be desirable.

Another drawback of the current tools is its disability to support a multi level annotation in a *structured* way. As there are no restrictions to add annotations, it is of course possible to add annotations on different levels, e.g. linguistic levels like syntax, semantic, morphology. Unfortuneatly these kind of annotation levels are not formally defined.

More text oriented approaches like the TEI[2] provide configurable rule sets and formalized descriptions (DTDs or RelaxNG) that guide the annotation process. In our approach we face the problem, that the annotation process is much more guided by the visual structure of the historical document and less by its textual structure.

We think that it is possible to combine these two approaches. The textual structure is often reflected in visual structure of a document. As such a TEI-guided annotation process could be realized by using the technique of *visual stand-off markup*. The visual annotations of the editor would then be interpreted either as TEI start-tags and end-tags, or as empty (target) elements, thus marking parts of the historical documents and providing reference to the exact visual position in the document. By this, visual regions of the document will be defined, which can then be structured in a hierarchical manner according to the TEI rules.

2.3. The user interface

We tried to make the user interaction with the annotation editor as simple as possible. It should very much behave in the same way a physical book would do (see image 1). To achieve this we incorporated touch technology. While touch control is already part of the current Windows 7 version it will be a central feature of the upcoming Windows 8. As such touch enabled displays are going to be a standard feature of the next generation of personal computers.

The annotation editor starts in its navigation mode. In order to make navigation simple and effective, a preview panel has been added. Here small thumbnails of the following

[2] http://www.tei-c.org/

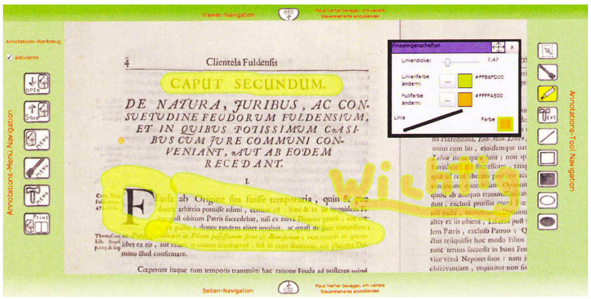


Figure 4: Free hand annotation

three pages are displayed. As the METS-file describes the structure of the document it is also possible to provide the user with a tree structured navigation menu. The editor also allows to define bookmarks an directly jump to specific pages. The pages of the "electronic book" can be flipped by simple drag gestures, zooming is achieved with a pinch gesture (see image 2), rotation with a simple two finger gesture.

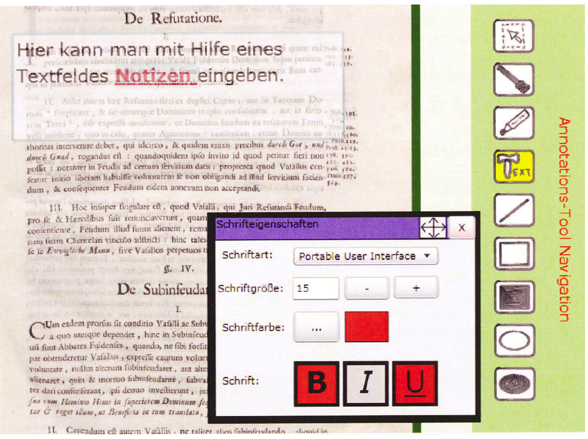
The touch interface is also functional in annotation mode. Here part of the digitized images can be marked, either in free hand mode (see image 3) or by using different geometrical shapes like rectangles, lines or circles. Commenting text boxes can be put onto the page at arbitrary positions and references between internal annotations and external sources can be established (see image 4).

The annotation mode provides a rather larger number of functions. In order to keep the interface simple, the annotation menu will only be displayed to user when needed. For the different annotation tools specific control boxes have been implemented making the easy to configure the markup process to the specific user demands. Text can be entered using physical an virtual keyboards. The annotations will be stored in unicode encoding, so special characters do not pose a problem for the system.

The annotation editor is implemented as a rich web client application running in the web browser. METS/MODS files are often distributed over the world wide web and the annotation editor is able to directly access these files. The security model of the application is exible enough to allow the user to store the results of the annotation process to the local computer.

One important feature of the application is its ability to efficiently work with extremly high resolution images. When working with his- torical documents it is crucially important to have access to even the smallest visual detail (see Schneider, 2009). Modern digital scanners provide a quite high resolution. In our case the file size of a single page image is larger than 200 MB. Even with broadband technology transmitting hundreds of such big images takes hours. As an effect many libraries only provide low resolution images on the internet.

Our editor is able to work with these large file in real time. This was achieved by using a Silverlight technology called deep zoom. Deep zoom images are created from high resolution images by automatically cutting the images into overlapping smaller tiles which will be trans- mitted, composed and displayed in the annotation editor.


Hier kann man mit Hilfe eines Textfeldes Notizen eingeben.

Schrifteigenschaften

Schriftart: Portable User Interface

Schriftgröße: 15

Schriftfarbe: 🟥

Schrift: B I U

Figure 5: Text box annotation

3. Conclusions and future work

The current version of the annotation editor is fully functional, but is still a research prototype. It is possible to interactively annotate historical documents, store the annotations as part of the METS/MODS file. The deep zoom function allows to work with high resolution doc- uments in real time over standard online connections.

Further developments of the system will focus on the better support of work flows for processing historical documents. So far the system is a single document editor. In order to better support comparative studies it would be nice to enable the parallel processing of multiple documents. Also it should be possible to define flexible annotations schemes for different working areas. This is a feature, that currently is missing from all available annotation tools.

4. References

Fabian Behrens and J.-T. Milde. 2006. The Eclipse Annotator: an extensible system for multimodal corpus creation. In *Proceedings of the third international conference on language resources and evaluation (LREC 2006), Genua*.

S. Bird and M. Liberman. 1999. A Formal Framework for Linguistic Annotation. *Speech Communication*, 33(1-2):23 –60.

D. Willems Peter Wittenburg Dan Broeder, F. Offenga. 2001. The IMDI Metadata set, its Tools and accessible Linguistic databases. In *IRCS Workshop*, Philadelphia.

Markus Geilfuss and J.-T. Milde. 2006. SAM - an annotation editor for parallel texts. In *Proceedings of the third international conference on language resources and evaluation (LREC 2006), Genua*.

Michael Kipp. 2001. Anvil - a generic annotation tool for multimodal dialogue. In *Proceedings of the 7th European Conference on Speech Communication and Technology (Eurospeech)*, pages 1367 – 1370.

METS. 2011. Official web site, February. http://www.loc.gov/standards/mets/.

J.-T. Milde and U. B. Gut. 2002a. A prosodic corpus of non-native speech. In B. Bel and I. Marlien, editors, *Proceedings of the Speech Prosody 2002 conference, 11-13*

April 2002. Aix-en-Provence: Laboratoire Parole et Langage, pages 503 – 506.

J.-T. Milde and U. B. Gut. 2002b. The TASX-environment: an XML-based toolset for time aligned speech corpora. In *Proceedings of the third international conference on language resources and evaluation (LREC 2002), Gran Canaria*.

MODS. 2011. Official web site, February. http://www.loc.gov/standards/mods/.

Felix Sasaki, Christian Wegener, Andreas Witt, Dieter Metzing, and Jens Pönninghaus. 2002. Co-reference annotation and resources: a multilingual corpus of typologically diverse languages. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC-2002), Las Palmas*. ELRA, Paris.

Thomas Schmidt. 2004. Transcribing and annotating spoken language with exmaralda. In *Proceedings of the LREC-Workshop on XML based richly annotated corpora, Lisbon 2004,*. ELRA, Paris.

Karin Schneider. 2009. *Palaeographie und Handschriftenkunde für Germanisten*. Niemeyer, Tübingen.

Bernd W. Seiler and Jan-Torsten Milde. 2003. *Goethes Werther, Bilder, Texte, Töne. Ein Literatur Kommentar auf CD-ROM*. C.C. Buchner Verlag, Bamberg.

Bernd W. Seiler and Jan-Torsten Milde. 2004. *Fontanes Effie Briest, Bilder, Texte, Töne. Ein Literatur Kommentar auf CD-ROM*. C.C. Buchner Verlag, Bamberg.

Tobias Sippel. 2011. Eine Anwendung zur touchgesteuerten Annotation elektronischer METS/MODS-Dokumente in Silverlight C#.

An Exploration of Language Identification Techniques for the Dutch Folktale Database

Dolf Trieschnigg[1], Djoerd Hiemstra[1], Mariët Theune[1], Franciska de Jong[1], Theo Meder[2]

[1]University of Twente, Enschede, the Netherlands
[2]Meertens Institute, Amsterdam, the Netherlands
{d.trieschnigg,d.hiemstra,m.theune,f.m.g.dejong}@utwente.nl,theo.meder@meertens.knaw.nl

Abstract

The Dutch Folktale Database contains fairy tales, traditional legends, urban legends, and jokes written in a large variety and combination of languages including (Middle and 17th century) Dutch, Frisian and a number of Dutch dialects. In this work we compare a number of approaches to automatic language identification for this collection. We show that in comparison to typical language identification tasks, classification performance for highly similar languages with little training data is low. The studied dataset consisting of over 39,000 documents in 16 languages and dialects is available on request for followup research.

1. Introduction

Since 1994 the Meertens Institute[1] in Amsterdam has been developing the Dutch folktale database, a large collection of folktales in primarily Dutch, Frisian, 17th century and Middle Dutch and a large variety of Dutch dialects (Meder, 2010). It does not only include fairy tales and traditional legends, but also riddles, jokes, contemporary legends and personal narratives. The material has been collected in the 19th, 20th and 21th centuries, and consists of stories from various periods, including the Middle Ages and the Renaissance. The database has an archival and a research function. It preserves an important part of the oral cultural heritage of the Netherlands and can be used for historical and contemporary comparative folk narrative studies. An online version has been available since 2004[2] and currently contains over 41,000 entries.

A rich amount of metadata has been assigned manually to the documents, including language, keywords, proper names and a summary (in standard Dutch). This metadata is very useful for retrieval and analysis, but its manual assignment is a slow and expensive process. As a result, the folktale database grows at a slow rate. The goal of the FACT (Folktales as Classifiable Text)[3] research project is to study methods to automatically annotate and classify folktales. Ideally, these techniques should aid editors of the folktale database and speed-up the annotation process. Language identification is the first challenge being addressed in the FACT project.

In this paper, we compare a number of automatic approaches to language identification for this collection. Based on the performance of these approaches we suggest directions for future work. The Dutch folktale database poses three challenges for automatic language identification. First, the folktales are written in a large number of similar languages. A total of 196 unique language combinations is present in the metadata; 92 unique (unmixed) language names are used[4]. For most of these languages no official spelling is available; the way words are spelled depends on the annotator who transcribed the oral narrative. As a result, documents in the same language may use a different spelling. For our experiments we have used a selection of 16 languages. Second, the language distribution in the collection is skewed: most of the documents are in Frisian and Standard (or modern) Dutch, but there is a long tail of smaller sets of documents in other languages. Consequently, for many languages only little training data is available to train a classifier. Third, documents in the collection can be multilingual. Most of the documents are monolingual, but some contain fragments in a different language. The length of these fragments ranges from a single passage or sentence to multiple paragraphs.

The contributions of this work are twofold. First, we present an analysis of multiple language identification methods on a challenging collection. Second, we make this collection available to the research community.

The overview of this paper is as follows. In Section 2 we briefly discuss related work. In Section 3 we describe the collection in more detail and outline the experimental setup. In Section 4 the results of the different classification methods are discussed followed by a discussion and conclusion in Section 5.

2. Related work

Early work on language learnability dates back to the 1960s (Gold, 1967). Since the 1990s language detection or language identification has become a well-studied natural language processing problem (Baldwin and Lui, 2010). For clean datasets, with only few and clearly separable languages, language identification is considered a solved problem (McNamee, 2005).

Recent research indicates, however, that language identification still poses challenging problems (Hughes et al., 2006), including: supporting minority languages, such as the dialects encountered in our collection; open class language identification, in such a way that a classifier is capable of indicating that no language could be accurately determined; support for multilingual documents; and classification at a finer level than the document level. Xia et al. (2009) and Baldwin and Lui (2010) also argue that language identification has not been solved for collections con-

[1] `http://www.meertens.knaw.nl`

[2] `http://www.verhalenbank.nl` (in Dutch only)

[3] `http://www.elab-oralculture.nl/FACT`

[4] Sometimes caused by an inconsistent naming convention

taining large numbers of languages. In this work we will focus on the capability of existing classifiers to deal with minority and very similar languages.

A large array of methods has been developed for tackling the problem of language identification: categorisation based on n-grams (Cavnar and Trenkle, 1994), words or stopwords (Damashek, 1995; Johnson, 1993), part-of-speech tags (Grefenstette, 1995), syntactic structure (Lins and Gonçalves, 2004), systems based on markov models (Dunning, 1994), SVMs and string kernel methods (Kruengkrai et al., 2005), and information theoretic similarity measures (Martins and Silva, 2005). An extensive overview of techniques is outside the scope of this paper. A more comprehensive overview can be found in Hughes et al. (2006) and Baldwin and Lui (2010). We limit our experiments to the method by Cavnar and Trenkle (1994) and a number of variations based on n-grams and words motivated by positive experimental results of (Baldwin and Lui, 2010).

3. Experimental setup

In the following subsections we describe the collection, investigated classification methods, and evaluation metrics in detail.

3.1. The collection

The complete folktale database[5] consists of over 41,000 documents. After filtering out documents with offensive content (sexual, racist, lese-majesty, etcetera) and copyrighted materials, 39,510 documents remain. From this collection we put all documents with a mixed language where at least one of the languages is Standard Dutch into a single language group labeled "Standard Dutch mixed". Documents in a language with fewer than 50 documents in that language in the collection are removed. This results in a collection of 39,003 documents in 16 different languages. Table 1 lists the 16 languages and their document frequencies. Note that the number of documents per language is strongly skewed: 79% of the collection is written in Frisian or Standard Dutch. The remaining 21% of the documents is distributed over the remaining fourteen languages. Also note that in comparison to previous work by Baldwin and Lui (2010), which uses collections between 1500 and 5000 documents, the collection is relatively large.

3.2. Classification methods

As a baseline classification method, we used the TextCat[6] implementation of the algorithm described by Cavnar and Trenkle (1994). The algorithm creates an n-gram profile for each language and performs classification by comparing each of the n-gram profiles to the n-gram profile of the text to classify. An out-of-place distance measure is used to compare the order of n-grams in the profile and the text. Following the methods investigated by Baldwin and Lui (2010) we used a number of classification methods based on nearest neighbour (NN) and nearest prototype (NP) in combination with the cosine similarity metric.

All tested classification methods use a supervised learning approach: classifications are based on a training set of manually labeled examples. The difference between NN and NP methods is the way the examples are stored. In the NP case, the examples of the same class are aggregated into a *prototype*, a single model representing the class. The prototype is constructed by summing the vectors of the examples. In the NN case, the examples are stored separately. During classification the class(es) of the nearest example(s) is/are returned. In our case we use the class of the first nearest neighbour (or prototype).

The documents are represented by vectors of the unit of analysis, containing the count of that unit. In the case of words, each unique word encountered in the collection forms one dimension of the vector. We use six different units of analysis: overlapping character n-grams of size 1 to 4, a combined representation of n-grams of length 1 to 4, and words (uninterrupted sequences of letters). The text is lowercased and punctuation is removed before features are extracted. The overlapping character n-grams are extracted by sliding a window of n characters over the text one character at a time. In case of the combined n-gram representation, this process is repeated four times (for n=1 to n=4). To reduce the complexity, we experiment with reducing the vector to a selection of 100, 500 and 1000 features. The selection of features is based on the most frequently used features per language appearing the training set. To be more precise: from each language the most frequent feature is taken until the desired number of features is reached. In our experiments we follow the approach described by Baldwin and Lui (2010). Alternatively, we could have used information gain to select the most informative features. We will consider this in future work.

3.3. Evaluation method

We evaluated the different approaches by means of stratified 10-fold cross-validation: the collection was split into

Language	Doc. count
Frisian	17,347
Standard Dutch	13,632
17th century Dutch	2,361
Standard Dutch mixed	1,538
Flemish	882
Gronings[1]	854
Noord-Brabants[1]	677
Middle Dutch	656
Liemers[1]	328
Waterlands[1]	153
Drents[1]	150
Gendts[1]	116
English	97
Overijssels[1]	80
Zeeuws[1]	68
Dordts[1]	64
Total (16 languages)	39,003

[1] Dutch dialects

Table 1: Collection statistics

[5] As of January 2012

[6] http://www.let.rug.nl/vannoord/TextCat

Language	Precision	Recall	F
Frisian	0.999	0.976	0.987
17th century Dutch	0.983	0.978	0.980
Middle Dutch	0.952	0.974	0.963
Liemers	0.861	0.909	0.884
Gronings	0.882	0.785	0.830
Standard Dutch	0.879	0.633	0.736
Gendts	0.942	0.560	0.703
Noord-Brabants	0.331	0.558	0.415
Zeeuws	0.692	0.265	0.383
Flemish	0.229	0.810	0.357
Dordts	0.207	0.609	0.310
Drents	0.196	0.707	0.307
English	0.112	0.887	0.199
Waterlands	0.091	0.824	0.163
Standard Dutch mixed	0.259	0.088	0.131
Overijssels	0.055	0.250	0.090
Macro average	0.542	0.676	0.527
Micro average	0.799	0.799	0.799

Table 2: Per-language classification performance for TextCat, sorted by descending F-score

10 stratified folds (preserving the proportion of languages in the whole collection). Each fold was used to test the method trained on the other nine folds.

As evaluation measures we use macro and micro averaged Precision, Recall and F-measure. The macro (or category) scores indicate the classification performance averaged over the languages, whereas the micro averaged scores indicate the average performance per document. For a particular language, precision is defined as the proportion of predictions in that language which is correct. Recall is the proportion of documents in that language that is correctly predicted. Note that for this classification task the micro average precision, recall and f-measure have the same value (hence the single column P/R/F in Table 4).

4. Results

4.1. TextCat baseline

Table 2 lists the classification performance of TextCat for the 16 languages in the collection. The contingency table in table 3 provides further information about the classification errors made. Its rows list the actual classes where its columns indicate the predicted classes indicated by the system. For example, the second row and first column indicates that 6 documents in Standard Dutch were incorrectly classified by TextCat as Frisian.

We can make the following observations. First, the classification performance of the largest language class (Frisian) is very good. The recall is very high (0.98) at almost perfect precision (0.999). Second, the classification performance of old Dutch languages (17th century Dutch and Middle Dutch) is also good (F-measure larger than 0.96). These languages can be distinguished well from modern Dutch and dialects. Third, the classification performance of the dialects is mixed. Some (Liemers, Gronings) perform relatively well, others (Waterlands, Overijssels) perform poorly. Still the highest F-measure (0.88) does not come close to typical performance scores, which range between 0.91 and 0.99 for the EuroGOV collection (Baldwin and Lui, 2010). Most of the dialects are mistaken for Standard Dutch and vice versa. Gronings shows strong overlap with Drents (both northern dialects); Zeeuws is frequently mistaken for Noord-Brabants, but not the other way around (both southern dialects). Fourth, it is striking that classification of English documents is so poor. Table 3 indicates that Standard Dutch and Standard Dutch mixed is frequently mistaken for English. One possible explanation is that English words or expressions are frequently borrowed in Dutch. It could also indicate that the annotation in the collection is inconsistent: the (Dutch) document contains an English expression but has been classified as Standard Dutch instead of Standard Dutch mixed.

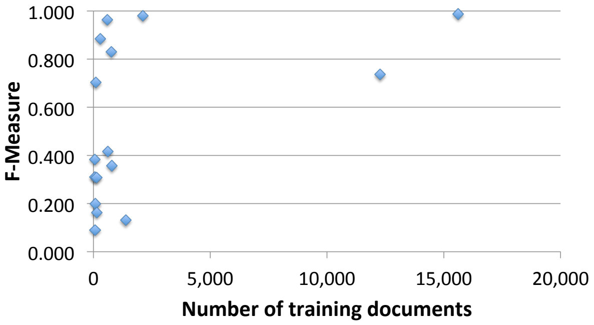


Figure 1: Amount of available training data and classification performance for TextCat

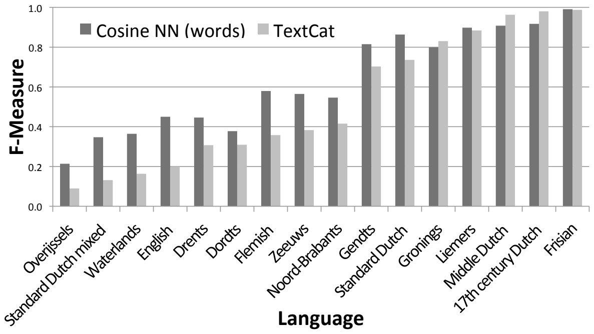


Figure 2: Per language classification performance: TextCat versus cosine (languages sorted according to classification performance of TextCat)

The micro average performance score (see Table 2) indicates a reasonable classification performance of TextCat, but this value has been strongly influenced by the strong performance on the largest language class. The macro averages illustrate that for many smaller languages classification performance is low. Figure 1 shows a scatter plot of the amount of training data available for a language and its classification score.

4.2. Variations of cosine distance

Table 4 summarises the classification performance of a number of variations on language identification systems. TextCat can be viewed as a variation of a nearest prototype system and is therefore in the left part of table.

Again, a number of observations can be made. First, TextCat performs better than all the cosine variants of the nearest prototype method (in terms of F-measure). All the nearest prototype variants based on cosine perform worse.

	Predicted															
Actual ↓	Frisian	Standard Dutch	17th century Dutch	Standard Dutch mixed	Flemish	Gronings	Noord-Brabants	Middle Dutch	Liemers	Waterlands	Drents	Gendts	English	Overijssels	Zeeuws	Dordts
Frisian	16928	106		5	31	27	78	3	5	48	53		7	48		8
Standard Dutch	6	8630	13	366	2065	12	446	6	20	1028	195	1	554	156	4	130
17th century Dutch		21	2308	3	13		2	12		1			1			
Standard Dutch mixed	1	695	9	135	194	15	165	7	4	131	32		92	48	3	7
Flemish		124	1	1	714		4		3	9			24	2		
Gronings		45		3	3	670	19	1		14	84	1		14		
Noord-Brabants		118	1	6	63	8	378	2	5	18	25		3	47	1	2
Middle Dutch		1	15		1			639								
Liemers		14			3		3		298		4	1		5		
Waterlands		15			9		2			126	1					
Drents		4		1		28	1				106			10		
Gendts		1			1		11		10	3	16	65		9		
English	3	4					1	1					86			2
Overijssels		21			7		8			6	18			20		
Zeeuws		6		1	4		23		1	4	5	1		5	18	
Dordts		11			5		2			4	1			2		39

Table 3: Contingency matrix for TextCat

		Nearest prototype				*Nearest neighbour*			
Character	# Features		Macro		Micro		Macro		Micro
n-grams		Precision	Recall	F	P/R/F	Precision	Recall	F	P/R/F
TextCat		**0.542**	0.676	**0.527**	**0.799**	-	-	-	-
Cosine									
n = 1	all (59)	0.234	0.489	0.243	0.498	0.404	0.419	0.407	0.781
n = 2	100	0.356	0.572	0.365	0.577	0.564	0.525	0.531	0.845
	500	0.405	0.598	0.410	0.597	0.629	0.562	0.579	0.864
	1000	0.406	0.599	0.410	0.598	0.631	0.564	0.581	0.865
	all (1,630)	0.406	0.599	0.410	0.598	0.631	0.564	0.581	0.865
n = 3	100	0.340	0.547	0.338	0.569	0.478	0.494	0.475	0.819
	500	0.451	0.628	0.449	0.630	0.606	0.565	0.561	0.855
	1000	0.484	0.635	0.475	0.630	0.628	0.583	0.582	0.863
	all (17,894)	0.503	0.643	0.490	0.631	0.664	0.598	0.606	0.874
n = 4	100	0.309	0.525	0.323	0.583	0.449	0.408	0.418	0.804
	500	0.375	0.632	0.400	0.637	0.588	0.521	0.540	0.852
	1000	0.376	0.654	0.409	0.641	0.621	0.543	0.568	0.864
	all (112,419)	0.403	**0.693**	0.442	0.656	**0.702**	0.584	0.624	0.886
n = 1...4	100	0.289	0.544	0.309	0.562	0.526	0.516	0.514	0.837
	500	0.354	0.607	0.382	0.638	0.585	0.564	0.567	0.866
	1000	0.372	0.624	0.401	0.658	0.611	0.582	0.588	0.874
	all (132,002)	0.400	0.650	0.431	0.687	0.669	0.601	0.624	**0.887**
words	100	0.369	0.650	0.394	0.643	0.474	0.490	0.475	0.828
	500	0.326	0.560	0.338	0.600	0.612	0.581	0.587	0.862
	1000	0.366	0.638	0.389	0.637	0.627	0.591	0.601	0.867
	all (174,180)	0.373	0.659	0.400	0.649	0.675	**0.609**	**0.630**	0.883

Table 4: Classification performance of evaluated systems

The nearest neighbour cosine variants perform similar or better than TextCat in terms of micro and macro F-measure. It should be noted, however, that these nearest neighbour approaches are far more expensive in terms of processing time and required storage than the method implemented by TextCat. Second, the cosine variants perform better with longer representations (longer n-gram windows or words) and with more features. Using all features performs best, but the selection of 1000 features closely approximates the scores based on all features. Figure 2 illustrates the difference between TextCat and the (NN) Cosine distance with word features: Cosine performs better on all languages, except Middle and 17th century Dutch, and Gronings.

5. Conclusions and future work

In this work we have investigated a number of language identification methods on a new and large collection of folktales in a variety and mix of languages. In comparison to other nearest prototype methods, the approach based on mixed n-grams proposed by Cavnar and Trenkle (1994) performs well. The results showed that a nearest neighbour approach using longer and more features performs even better.

Compared to other language identification tasks carried out by Baldwin and Lui (2010), the classification results stay behind. Baldwin and Lui (2010) report a maximum macro F-measure of 0.729 for a skewed collection containing 67 languages. With similar methods, we achieve only 0.630, for a collection with fewer languages. These results indicate that this collection indeed poses a challenge for language identification. The collection therefore is a valuable resource for future language identification research. The collection is available on request (users are required to sign a license agreement).

An important note has to be made on the consistency of the language annotations in the collections. The folktales in the database have been gathered and annotated (in a free text field) by more than 50 people. It is an open question whether these editors have used the same method for labelling the language of a document; some might have annotated a document with Standard Dutch, where another would have labeled it as a mix of Standard Dutch and another language. This might explain why the automatic methods cannot discriminate between these classes.

Our future work will focus on the following aspects of language identification. First, we intend to focus on multilingual document detection. Almost 10% of the documents in the complete collection contains multiple languages. Therefore, it would be useful to detect languages at the sentence level. Second, it would be useful to assign a level of certainty to the detected language. In the work described in this paper we view the task as a closed classification problem with a fixed number of languages. Especially for the long tail of documents in minority languages it would be useful to indicate if no known language was confidently determined. Third, since the language identification system is intended to be used in a semi-automatic setting, it is useful to have a mechanism to present proof for the detected language. Especially when the annotator has no in-depth knowledge of the different languages this would be useful. This could be achieved, for example, by showing sentences from the suggested language(s) similar to the sentence under classification. Fourth and finally, since classification performance is still relatively low, we intend to investigate how contextual information can be used to improve classification performance. In the line of recent work from Carter et al. (2013), who improved the language identification of Twitter messages by incorporating classification features based on for example language of the blogger and language of the document linked to, we could introduce additional features for this particular domain. One can think of features based on the date, source, and place of narrative of the folktale. Or a feature based on the geographical locations encountered in the text. In addition, it might be possible to incorporate knowledge from dialect lexicons to improve classification.

6. Acknowledgements

This work has been carried out within the Folktales as Classifiable Texts (FACT) project, which is part of the CATCH programme funded by the Netherlands Organisation for Scientific Research (NWO).

7. References

T. Baldwin and M. Lui. 2010. Language identification: The long and the short of the matter. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the ACL*, pages 229—237, Los Angeles, California, USA.

S. Carter, W. Weerkamp, and E. Tsagkias. 2013. Microblog language identification: Overcoming the limitations of short, unedited and idiomatic text. *Language Resources and Evaluation Journal*. To appear.

W.B. Cavnar and J.M. Trenkle. 1994. N-gram-based text categorization. In *Proceedings of the Third Symposium on Document Analysis and Information Retrieval*, Las Vegas, USA.

M. Damashek. 1995. Gauging similarity with n-grams: Language-independent categorization of text. *Science*, 267(5199):843.

T. Dunning. 1994. Statistical identification of language. *Computing Research Laboratory Technical Memo MCCS*, pages 94–273.

E.M. Gold. 1967. Language identification in the limit. *Information and control*, 10(5):447–474.

G. Grefenstette. 1995. Comparing two language identification schemes. In *JADT 1995, 3rd International Conference on Statistical Analysis of Textual Data*, Rome, Italy.

B. Hughes, T. Baldwin, S. Bird, J. Nicholson, and A. MacKinlay. 2006. Reconsidering language identification for written language resources. In *Proc. International Conference on Language Resources and Evaluation*, pages 485–488.

S. Johnson. 1993. Solving the problem of language recognition. Technical report, Technical report, School of Computer Studies, University of Leeds.

C. Kruengkrai, P. Srichaivattana, V. Sornlertlamvanich, and H. Isahara. 2005. Language identification based on string kernels. In *Communications and Information Technology, 2005. ISCIT 2005. IEEE International Symposium on*, volume 2, pages 926–929. IEEE.

R.D. Lins and P. Gonçalves. 2004. Automatic language identification of written texts. In *Proceedings of the 2004 ACM symposium on Applied computing*, pages 1128–1133. ACM.

B. Martins and M.J. Silva. 2005. Language identification in web pages. In *Proceedings of the 2005 ACM symposium on Applied computing*, pages 764–768. ACM.

P. McNamee. 2005. Language identification: a solved problem suitable for undergraduate instruction. *J. Comput. Small Coll.*, 20:94–101, February.

T. Meder. 2010. From a Dutch folktale database towards an international folktale database. *Fabula*, 51(1-2):6–22.

F. Xia, W.D. Lewis, and H. Poon. 2009. Language id in the context of harvesting language data off the web. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 870–878. Association for Computational Linguistics.