# The *eIdentity* Text Exploration Workbench

**Fritz Kliche[1], André Blessing[2], Ulrich Heid[1], Jonathan Sonntag[3]**

[1]IWiSt, Universität Hildesheim, [2]IMS, Universität Stuttgart, [3]Linguistics Department, Universität Potsdam
{kliche,heid}@uni-hildesheim.de, andre.blessing@ims.uni-stuttgart.de, jonathan.sonntag@yahoo.de

**Abstract**

We work on tools to explore text contents and metadata of newspaper articles as provided by news archives. Our tool components are being integrated into an "Exploration Workbench" for Digital Humanities researchers. Next to the conversion of different data formats and character encodings, a prominent feature of our design is its "Wizard" function for corpus building: Researchers import raw data and define patterns to extract text contents and metadata. The Workbench also comprises different tools for data cleaning. These include filtering of off-topic articles, duplicates and near-duplicates, corrupted and empty articles. We currently work on ca. 860.000 newspaper articles from different media archives, provided in different data formats. We index the data with state-of-the-art systems to allow for large scale information retrieval. We extract metadata on publishing dates, author names, newspaper sections, etc., and split articles into segments such as headlines, subtitles, paragraphs, etc. After cleaning the data and compiling a thematically homogeneous corpus, the sample can be used for quantitative analyses which are not affected by noise. Users can retrieve sets of articles on different topics, issues or otherwise defined research questions ("subcorpora") and investigate quantitatively their media attention on the timeline ("Issue Cycles").

**Keywords:** Corpus Building, Metadata, Digital Humanities

## 1. Introduction

A vast amount of text data from different sources is available for research in the Digital Humanities. Usually, text contents is embedded in structured (meta-) data of some kind. Digital Humanities-oriented work on the texts, including full text search, filtering, or the use of NLP tools, requires preprocessing steps to make the content accessible and to exploit the metadata. These steps are usually achieved through implementing scripts to capture the data structures and to import the text contents into a database. We work on a tool suite which can be used to perform these steps without implementing scripts. Users upload a snippet of their raw data into a *Wizard*, where they define patterns to extract text contents and metadata. They can generalize these patterns over their data set to import the data into a repository, where they are accessible for data cleaning, retrieval of topically defined subcorpora, and full text search. We integrate these tools into our *Exploration Workbench*. We also integrate existing NLP tools into the Workbench, including POS tagging, syntax parsing, and named entity recognition, which are used for defining the patterns for the extraction of text contents and metadata.

We develop our tools as part of the Digital Humanities project *eIdentity*. The data stem from several news archives which use different representation formats for both text content and metadata. Coverage and representation of metadata vary between different archives, but a core set is shared, e.g. publishing date, author and newspaper section. The Workbench allows for building topically defined subcorpora on the cleaned sample. In our current sample, political scientists empirically investigate different ways to evoke *collective identities*, such as different readings of *religious*, *national*, *European*, etc., identities. These identity notions are verbalized in expressions like *we as Europeans*, *common market*, or *our common history*. Political scientists are interested in the abstract, generalized identity notions and use word lists with terminology which is indicative of the different *identity* concepts to extract sentences or text passages containing these indicator items, in order to compile subsets of the sample according to a given type of identity notion. The clean sample allows for quantitative comparisons of these subsets which are more reliable than comparisons that might be affected by noise in the sample.

The central task of the Workbench is to help reproduce the "meaning" of the data structures of raw text data as provided by newspaper archives. To our knowledge, there are no tools available to infer text structure and metadata from the data structures of heterogeneously structured text data, in order to make the text contents accessible for NLP tools. Existing corpus builders which make text data accessible for NLP methods, require a fixed data format for import. Corpus query tools such as *COCA* (Corpus of Contemporary American English) don't allow for the definition of user-defined sets of metadata (Davies, 2010); the *Metadata Editor* of TextGrid (Kerzel et al., 2009) allows for the manual annotation of user-defined metadata, but doesn't include methods to infer metadata or text-structural elements from the structure of raw data. Our project partners reported difficulties using the text mining tool *WordStat* of *Provalis*[1] for this purpose. Here, the idea came up for (1) a tool for the conversion of the raw newspaper data of our current collection, (2) which is generic enough to be used for text data of future projects.

Section 2 gives a user-oriented overview of the functions of the Workbench. The *Wizard* function and the

[1]http://provalisresearch.com/products/content-analysis-software/

*rule language* to extract text contents and metadata are laid out. Next, the functions for checking the imported data for consistency, for cleaning the sample, and for the subsequent work on the text contents are described. In section 3, the description of a use case of our current *eIdentity* sample exemplifies these functions and gives details of their implementation.

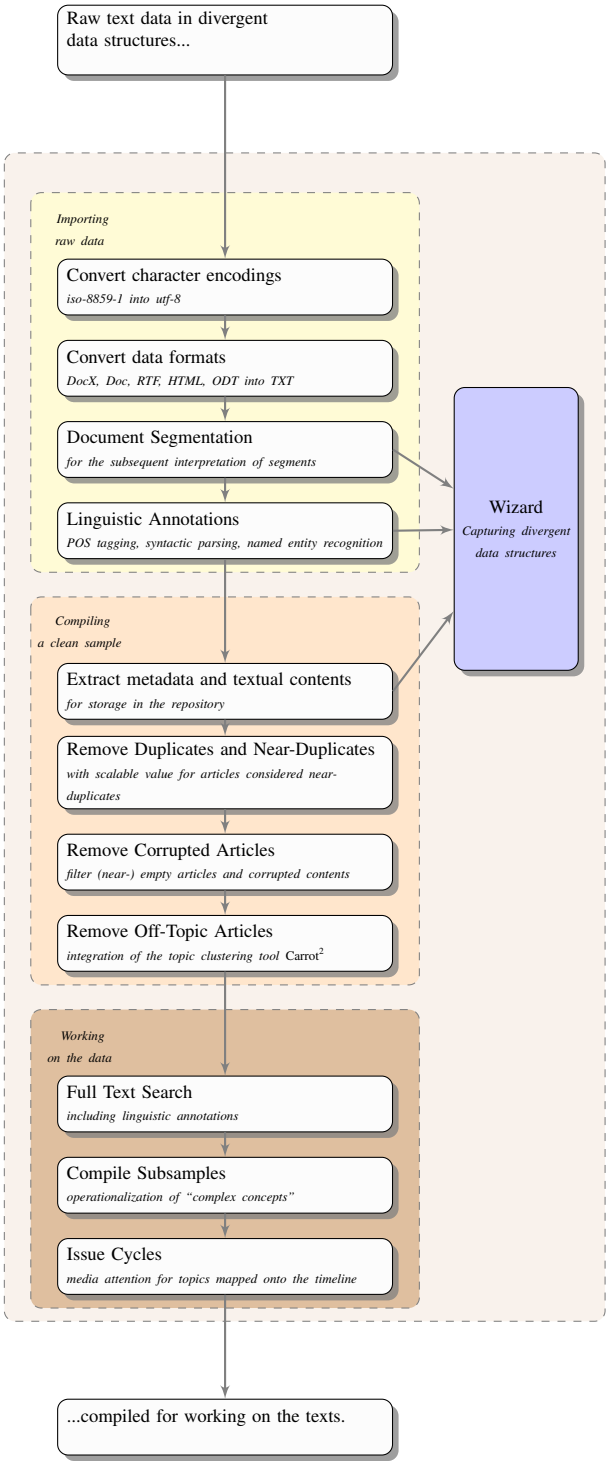


Figure 1: Overview on the functions of the *eIdentity Exploration Workbench*.

## 2. Functions of the Workbench: the user view

Figure 1 sketches the processing chain of the *Exploration Workbench*. Users upload raw text data in different data formats (DocX, Doc, RTF, HTML, ODT (Open Office), and TXT) and character encodings (iso-8859-1 and utf-8). The data are converted into utf-8 plain text on the fly. The imported documents are sliced into text segments. In the current setting, segments are lines separated by a newline. In a future version of the Workbench, segment delimiters can be defined by the user.

The following functions of the *Exploration Workbench* are discussed in the remainder of this section.

### 2.1. Wizard to capture data structures

We develop a *Wizard* to capture the data structures of raw data and to import the data into a repository. The Wizard function can be used in two stages of the data import process: (1) Users slice raw text data files into separate *articles*, i.e. chunks of raw data containing text content and metadata; (2) *articles* are further segmented to extract the text content and metadata.

*Extraction of articles*

Users upload a file of their data set which is displayed in a preview window. Figure 2 shows a snippet of the *eIdentity* sample. In this sample, the raw data consist of text files built up from several articles (~100 articles per file), each article in turn containing metadata and text content. Next, users define patterns ("segmentation rules") which mark boundaries between articles. In figure 2, the end of an article is indicated by *"Document abbeik..."*. A closer inspection of more articles from the same archive shows that the end of each article is in this case marked by (and thus inferable from) a sequence of the token 'Document' and a string of letters and numbers. This pattern is captured by a *segmentation rule*: We define a 'rule language' to express segmentation rules (patterns); the segmentation rule in the given example is '[begin] Document [word] [end]'.

The rule language has to be a trade-off between expressiveness and easy usability, as we don't want to substitute the implementation of programming language scripts with a 'rule language' with its own (probably similar) complexity. It is not sufficient to offer regular expressions. Therefore, the rule language uses a fixed set of categories. The categories for slicing articles out of raw data include the following:

- Definitions of fixed tokens, such as 'Document' in the above example;
- Abstractions over types of tokens, such as 'Letters'; 'Numerals'; 'any Characters'; 'Uppercase Letters', etc.;

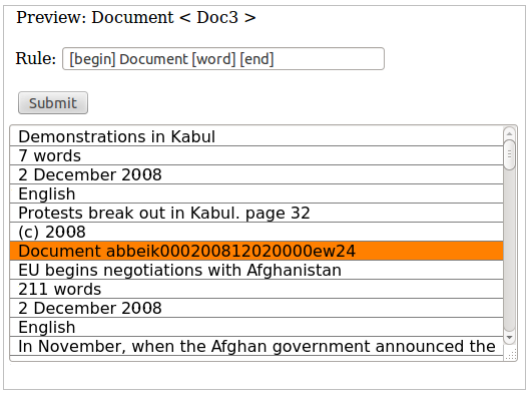


Figure 2: Wizard to segment raw text files into *articles*.

- Context rules, limited to the next and the following segment, e.g. constraints such as 'the following text segment must be empty';
- Limitation on the length of the segment.

When developing and testing segmentation rules, users can first apply their segmentation rule to the file in the preview window, where matching text segments are highlighted. Once checked and validated, the segmentation rules can be applied to import all raw data into the repository. The raw data are split up according to the segmentation rules, and each raw *article* is imported into the repository, labelled with an ID.

*Extraction of text contents and metadata*

Patterns similar to the above ones can also be applied to extract metadata and text passages from the raw *articles*. Again, the patterns are applied to "text segments", thus to newline-separated lines of raw data. At this step of processing, the articles are stored in the repository, where the data are accessible for further linguistic processing (POS tagging, syntactic parsing, named entity recognition). Additionally, users can include their own terminology, e.g., a list of month names for the extraction of dates. Alongside extraction, also the annotation of text segments is possible at this stage. Annotations may make use of two further categorization types:

- Abstractions over types of tokens using linguistic classifications, such as part of speech or named-entity labels;
- User-defined or pre-defined terminology, such as lists of 'persons names', 'month names', etc.

Figure 3 shows the segmentation of an article using the rule language. Users define patterns to extract metadata and text contents. As in the case of the articles extracted from a data file, also the metadata and the text contents can be stored in the repository. Text contents is converted into XML.

### 2.2. Checks for consistency of the sample

The pattern-based text segmentation (for segmenting raw data and subsequently articles) presupposes consistency of

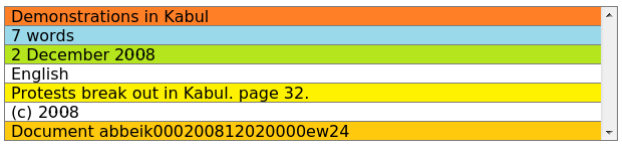


Figure 3: Extraction of text contents and metadata from raw *articles*.

the data structures, at least to some extent. The concept underlying the Wizard is to define rules based on an inspection of a snippet of the data, and on the subsequent generalization to the full data set. We integrated means to check the data that have been imported into the repository for consistency.

For the segmentation of raw data into articles (i.e. the first import step), the number of articles that have been sliced out of one file is displayed. Strong deviations from average numbers indicate erroneous data imports.

The Workbench also contains checks for the segmentation of articles into text contents and metadata (i.e. the second import step). Users can inspect the extracted text contents and the metadata. They find groups of articles that were segmented in the same way, e.g. a group of articles for which text content and metadata on 'publishing date', 'author' and 'headline' have been extracted, but no further metadata; or articles for which 'author' indications begin with *By our correspondent*. When finding that the article segmentation is correct for a given group, users can generalize over this group and mark the corresponding articles with a flag indicating 'correct import'. Subsequently, they can check the remaining set of articles for consistency. Thus, they can reduce the set of articles which have to be checked for consistency. Using these flags, we want to introduce a means to check data on a large scale. It is evident, however, that the inspection of some data entries and the subsequent generalization only allow for approximative results and cannot replace exact measurements.

### 2.3. Cleaning from noise

Users can remove empty and almost empty articles (e.g. captions misclassified as articles). The Workbench assists users in interactively defining the appropriate length of articles to be kept. Another task of data cleaning is to detect corrupted articles. Heuristics can be used to detect irregularities in the data set (high percentage of special characters or numbers, unusual characteristics of metadata, etc.). Manual checking of removal candidates is supported by the Workbench, as it displays potentially corrupted articles to the user, who can adjust the settings for article filtering. (Near) duplicate articles are detected by the tools. Duplicates are not deleted from the repository, but flagged (as there may be research questions, e.g. about the reuse and spreading of items circulated by news agencies, which require a more detailed analysis of such cases to be answered satisfactorily).

### 2.4. Removing off-topic articles

The Workbench includes a topic clustering tool. It clusters the imported articles into thematic groups and generates a label for each cluster. Users can inspect the articles of a cluster. When finding that a cluster is not relevant for a sample, they set a flag 'off-topic' for the articles of this cluster.

### 2.5. Full Text Search

The imported text contents is accessible to full text search. Search functions include the search for word sequences and for words cooccurring (not necessarily adjacently) in sentences. Linguistic annotations can be used in the queries. Figure 4 shows results for a query "wir vvfin" on a German newspaper sample. The query uses the token "wir" ('we') and the POS tag "vvfin" ('finite verb').

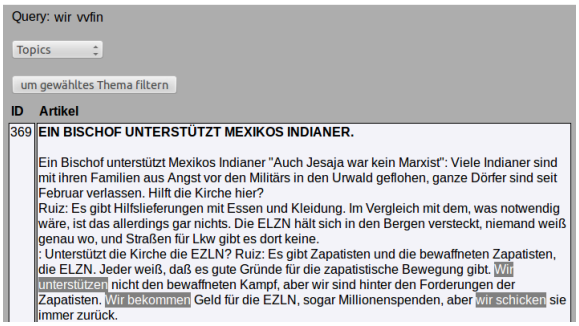


Figure 4: Full text search using linguistic abstractions: Search results for the query "wir vvfin" using POS tags.

### 2.6. Operationalizing complex concepts

One goal of the *Exploration Workbench* is to provide a means for the quantitative comparison of thematically defined "subcorpora". Users can compile lists of keywords and phrases which are indicative for a topic, and retrieve the corresponding articles from the sample. Deploying the metadata on publishing dates, users can map the amount of retrieved articles per year, per month, per week, or per day onto the timeline. This mapping shows the media attention for a concept ("Issue Cycles") over time (also cf. figure 6 for an example from our current sample). By these means, users can investigate the interplay between different positions or opinions ("concepts"), and peaks and losses of media attention.

## 3. Exploring the *eIdentity* corpus: Use case and implementation

We work on a large trilingual (English, French, German) collection of newspaper articles (~860.000 articles). We collected data from 12 newspapers from 6 countries (DE, AT, FR, GB, USA, Ireland) covering the sampling period 1990 – 2012. The data stem from five licensed digital media archives. They were collected using lists of keywords. We gathered articles from the domain of foreign politics, which deal with *military interventions in international crisis situations*. The sample for each newspaper varies between 7.231 articles and 135.778 articles.

The data were delivered in RTF, HTML, and plain text (TXT without markup) formats, and contained articles in iso-8859-1 and utf-8 character encodings. The articles were delivered in files containing several (~100) articles, in their own data structure for each media archive, wrapped with metadata. These metadata include: publishing date, author, newspaper section, subsection, word count, page, article id; additionally, the following text segments were given as metadata: headline, teaser, subtitle. The sets of metadata differ between media archives and sometimes even between articles; e.g. "authors" or "subtitles" are not given for each article. For *eIdentity*, we first processed the sample by implementing scripts, and made these scripts accessible and manageable in a browser GUI, which led to the *Exploration Workbench*. As a first step, the files were converted into plain text (TXT) in utf-8. Next to converting RTF and HTML, we included into the Workbench further converters that translate DocX, Doc, and ODT into TXT. Additionally, documents are checked for HTML special character encodings. The extracted text contents and metadata are stored in a repository (SQL database). On the basis of regular expressions and of our experiences wrt. required functions, we proposed the elements of the "rule language" for the Wizard. The results of the script-based extraction of metadata and contents serve as a gold standard for testing the functions of the Wizard. The functions of the Wizard were implemented after the current sample was compiled.

An indication of the *author* of an article was in some cases given in the metadata, and in others, it was contained in the text body. We used anchor words like 'correspondent', the position in the text, or the lengths of segments as indicators for author indications in the text content. It turned out to be useful to integrate a list of first names, particularly in order to distinguish authors from subtitles. The anchor items and the list of first names were integrated into the rule language. Publishing dates in the conventional formats of GE, EN or FR were converted into a standardized representation (yyyy-mm-dd).

We integrated Solr[2] to offer full text search on the data. Solr provides efficient search functions, also for data on a large scale. Solr allows for the integration of full text search, using different types of queries. These include (1) single term queries ("LREC"), (2) phrase queries ("LREC 2014"), (3) proximity searches (two search terms in a defined maximal distance, e.g. "Language + Conference" matching "Language Resources and Evaluation Conference"). We combine UIMA (Ferrucci and Lally, 2004) with Solr. In the UIMA framework, existing NLP methods can be integrated as web services. In this way, we combine NLP functionalities accessible via UIMA with the rapidity of Solr. The CLARIN-ERIC infrastructure provides several tools as web services to process natural

[2]http://lucene.apache.org/solr

language text[3]. These tools are registered in the Virtual Language observatory (VLO). The tools listed below are used; the PID gives the link to the CMDI description:

- *Tokenizer*:
  Tokenizer and sentence boundary detector for English, French and German (Schmid, 2009)
  PID: http://hdl.handle.net/11858/00-247C-0000-0007-3736-B
- *TreeTagger*:
  Part of speech tagging for English, French and German (Schmid, 1995)
  PID: http://hdl.handle.net/11858/00-247C-0000-0022-D906-1
- *RFTagger*:
  Part of speech tagging for English, French and German, using a fine-grained POS tagset (Schmid and Laws, 2008)
  PID: http://hdl.handle.net/11858/00-247C-0000-0007-3735-D
- *German NER*:
  German Named Entity Recognizer, based on Stanford NLP (Faruqui and Padó, 2010)
  PID: http://hdl.handle.net/11858/00-247C-0000-0022-DDA1-3
- *Stuttgart Dependency Parser*:
  Mate tools dependency parser (Bohnet, 2010)
  PID: http://hdl.handle.net/11858/00-247C-0000-0007-3734-F

The above described tools can also be accessed by WebLicht (Hinrichs et al., 2010), but for large scale applications like our *Exploration Workbench* (cf. the numbers of news articles being dealt with in *eIdentity*), a direct interaction with the tools is more convenient. The WebLichtWiki (http://weblicht.sfs.uni-tuebingen.de/weblichtwiki/) provides a developer manual which describes how to use the WLFXB library which allows for interacting with WebLicht's TCF format and for interacting with the RESTStyle web services of CLARIN.

For topic clustering, we integrated the clustering tool 'Carrot[2]' into Solr[4]. It clusters a sample into *n* clusters, and generates a label for each cluster. Currently, *n* is set to 20.

Based on the metadata indicating publishing dates, we determine the amount of articles per year, per month, per week, and per day. If we find time periods where no or few articles have been gathered, we "re-sample" to fill the gaps.

We filter duplicate and near-duplicate articles. In *eIdentity*, we only consider articles with identical publishing days to be (near-) duplicates; this assumption reduces the runtime of our scripts drastically. Duplicated and near-duplicated articles may enter a sample for several reasons:

- Newspaper publishers upload an article twice on the same day (duplicate articles) or make slight corrections in the course of the day and publish the article afresh (near-duplicates);
- We may have, by error, collected articles from the same newspaper twice, from different media archives;
- Re-Sampling: For periods in the sample where few articles had been collected, "re-sampling" may have led to duplicates.

For duplicate detection, we use the methods of fingerprinting and shingling as outlined in (Manning et al., 2008). For fingerprinting, we reduce the articles to the 12 least frequent letters contained in the sample. We produce 5-gram shingles from the reduced articles (5-grams based on letters). Identical 5-grams are counted, resulting in a hash for each document, containing the document's 5-grams with their frequencies. Next, the similarity of each article pair is calculated, based on the counting of 5-grams which occurred in both documents. We used cosine similarity as the standard way of quantifying the similarity of two documents (Manning et al., 2008, pg. 111). Be $\vec{X}$ the 5-grams of document A and $\vec{Y}$ the 5-grams of document B, i.e. hashes with the 5-grams and their frequencies. We calculate cosine similarity by dividing the inner product of the two vectors, i.e. the frequency values of common 5-grams, by the Euclidean lengths of the vectors, i.e. by the product of the "summed up" and normalized frequency values:

$$sim_{cos}(A,B) = \frac{\vec{X}\cdot\vec{Y}}{|\vec{X}||\vec{Y}|} = \frac{\sum_{i=1}^{M} x_i y_i}{\sqrt{\sum_{i=1}^{M} \vec{X_i}^2}\cdot\sqrt{\sum_{i=1}^{M} \vec{Y_i}^2}}$$

This yields a similarity score for each pair of documents, ranging from 0 (no similarity) to 1 (identical articles). Users can set a threshold value on the similarity score to determine when two articles are to be considered near-duplicates. We chose 0.6 as an appropriate threshold.

Figure 5 shows an example of near-duplicates. It is a snippet from a French article which was slightly changed and published afresh. The differences are indicated in bold text in this figure; they are not marked by the tools. A similarity score of ~0.65 was computed.

We also filter corrupted articles. We do so by setting flags which mark articles, instead of deleting articles from the sample. We set a flag "Empty" for empty articles, i.e. articles <9 characters, and a flag "Stub" for near-empty articles, i.e. articles <121 characters. For example, graphics and photos may enter the sample as separate articles, with the caption as their only text content. We set a flag "Num" for articles where more than 6 % of all characters are digits. We set a flag "Chars" for articles where more than 20 % of the characters are non-alphabetic. The flags "Num" and "Chars" are easy to implement, but yield good results. They filter articles which don't contain running

[3] http://de.clarin.eu/de/
[4] http://carrot2.org

FILE=k92.txt ID=1077
FILE=k98.txt ID=1103
Similarity:
0.645523532251084

« Il est temps que la Serbie affronte le passé », a réagi le premier ministre croate Ivo Sanader à Vukovar, où avaient lieu les cérémonies du 17e anniversaire de la prise de la ville par les forces yougoslaves et serbes après trois mois de siège. **Les autorités serbes ont indiqué hier qu'elles porteront plainte à leur tour, contre la Croatie pour crimes de guerre et nettoyage ethnique, en réponse aux accusations de Zagreb**.
Lors des audiences en mai, Belgrade avait argué que la Serbie ne pouvait endosser les responsabilités légales de l'ancienne République fédérative de Yougoslavie (RFY), partie en guerre, aux côtés des sécessionnistes serbes de Croatie, contre Zagreb qui venait de déclarer son indépendance en 1991. Mais les juges ont rejeté cet argument. Mme Higgins a rappelé que l'actuelle Serbie avait elle-même entamé des procédures devant la CIJ contre dix États qui avaient bombardé la **Serbie** en 1999. La Croatie s'attend à l'ouverture des procédures d'ici à trois ans.

« Il est temps que la Serbie affronte le passé », a réagi le premier ministre croate Ivo Sanader à Vukovar, où avaient lieu les cérémonies du 17e anniversaire de la prise de la ville par les forces yougoslaves et serbes après trois mois de siège. **La ministre serbe de la Justice, Snezana Malovic, a indiqué que son gouvernement pourrait porter plainte à son tour, tout en insistant sur la nécessité pour la région d'atteindre la « réconciliation »**.
Lors des audiences en mai, Belgrade avait argué que la Serbie ne pouvait endosser les responsabilités légales de l'ancienne République fédérative de Yougoslavie (RFY), partie en guerre, aux côtés des sécessionnistes serbes de Croatie, contre Zagreb qui venait de déclarer son indépendance en 1991. Mais les juges ont rejeté cet argument. Mme Higgins a rappelé que l'actuelle Serbie avait elle-même entamé des procédures devant la CIJ contre dix États qui avaient bombardé la **RFY** en 1999. La Croatie s'attend à l'ouverture des procédures d'ici à trois ans.

Figure 5: Example for near-duplicates. An article was slightly changed and published afresh.

text, but, e.g. stock exchange charts or cinema programs, and they filter corrupted articles without readable content. Finally, duplicate articles are marked with a flag as well. For a pair of duplicates, one article is marked with a flag, the other remains unmarked. If more than two articles are (near-) identical, only one article remains unmarked.

In our experiments, cleaning from noise shrank the sample as follows: After completion of the sampling period, the sample contained 863.034 articles. 13.805 articles were marked with a flag concerning formal criteria: 7.658 articles were marked with "Num", 2.248 articles were marked with "Chars". For 2.475 articles, the flag "Stub" was set, 1.424 articles were marked as "Empty". Articles could contain several flags. The text cleaning procedure which most importantly reduced the size of our sample was the removal of duplicates. 107.193 articles were marked as duplicates. In total, 112.901 articles and thus 13.08 % of the sample were removed by our filtering steps.

The tools of the Workbench are to a large extent language-independent. The Wizard function (except for the integration of NLP tools), the filtering steps and the removal of (near-) duplicates based on n-gram shingling are language-independent. As all NLP tools are integrated as web services, the tools can be adapted to further language resources. We integrate language-dependent tools (POS tagger, parser, named entity recognition) for English, German and French. The topic clustering tool *Carrot*[2] is also available multilingually.

On the clean corpus, political scientists have started to conduct quantitative research. In *eIdentity*, we investigate different "collective identities" and retrieve, for example, (1) the set of articles expressing a "common European identity", or (2) the set of articles appealing to a "common religious identity". We work on enhancing the functions for compiling topic-related "subcorpora" from a given sample and to operationalize "complex concepts". For a discussion on how we intend to capture these "complex concepts", see (Blessing et al., 2013). Currently, the political scientists gather keywords and key phrases which are indicative for a concept. These concepts include

- political actors:
  "Europeans", "Americans", "the European Union", "ministry", etc.;
- appeals to religion:
  "our protestant faith", "believe in god", etc.;
- appeals to democracy:
  "civic engagement", "human rights", "free and fair elections", etc.

Sets of articles on these topics can be mapped onto the timeline ("Issue Cycles"). Figure 6 depicts a snippet of the results for articles containing "appeals to democracy" from one of our newspapers. The percentage of these articles per month is given, measured on the cleaned sample. We plotted the results from the Workbench onto a graph. In the example, it shows the media attention for the topic in the time period 2003 – 2005 on a monthly basis.

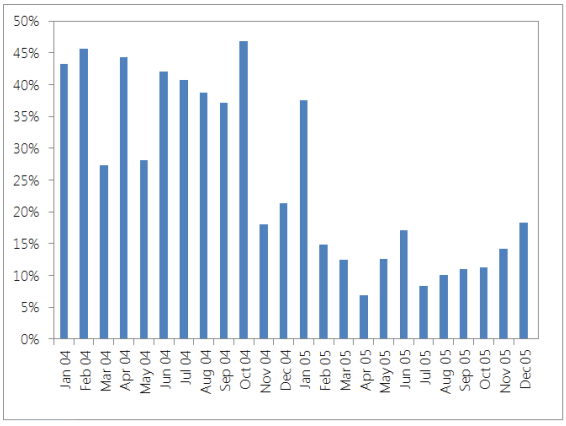


Figure 6: Percentage of articles per month of a sample for a newspaper expressing the concept of *democracy* in a period of time ("Issue Cylce").

## 4. Conclusion and future work

We outlined the *Exploration Workbench* which is being developed in the current Digital Humanities project

*eIdentity*. Researchers from the political sciences aim at quantitative investigations on large samples of newspaper articles. Our colleagues collect raw newspaper data from different sources and in different data formats. The task of the *Exploration Workbench* is to convert the raw data into a consistent, clean and reliable sample.

The processing pipeline starts with conversions of data formats (DocX, Doc, HTML, RTF, ODT) and character encodings (iso-8859-1 and utf-8) into utf-8 plain text (TXT). The media archives embed the text contents in different data structures and use different representations for metadata. Using the *Wizard* function, users can capture these data structures, without the need to implement scripts. Users upload the raw data and define patterns to slice items ("articles") out of the raw data, which are stored in a repository. In a second step, they extract text contents and metadata from the *articles*. In the repository, the data are indexed using Solr, and they are linguistically annotated. To this end, we use existing NLP tools which are integrated as web services from the CLARIN infrastructure. The Workbench includes methods for cleaning and filtering the sample, e.g. the removal of empty, near-empty, and corrupted articles, off-topic articles, duplicates, and near-duplicates. Finally, we described how users can compile thematical 'subcorpora' from the sample. By use of the metadata on publishing dates of the articles, the amount of articles of a subcorpus in a period of time can be mapped onto a timeline. This mapping can be used to analyse the media attention for a topic over time (referred to as "Issue Cycles").

We will continue our work on retrieving the sets of articles in which different concepts of *identities* are expressed. Currently, we use terminology lists to compile these subsets. We have included NLP methods into our setting, and continue to work on methods which more reliably retrieve articles on the intended concepts from the political sciences. We will deploy the Wizard functionality on data from further data sources. New use cases will lead to improvements of the "rule language" and will improve the genericity of the Workbench, beyond the scope of our current sample.

## 5. Acknowledgements

*eIdentity* is funded by the German Ministry of Education and Research (BMBF Grant No. 01UG1234) within the *eHumanities* program.

## 6. References

Blessing, André, Sonntag, Jonathan, Kliche, Fritz, Heid, Ulrich, Kuhn, Jonas, and Stede, Manfred. (2013). Towards a tool for interactive concept building for large scale analysis in the humanities. In *Proceedings of the 7th Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*, pages 55–64, Sofia, Bulgaria. Association for Computational Linguistics.

Bohnet, Bernd. (2010). Top accuracy and fast dependency parsing is not a contradiction. In *Proceedings of the 23rd International Conference on Computational*, pages 89–97.

Davies, Mark. (2010). The corpus of contemporary American English as the first reliable monitor corpus of English. *LLC*, 25(4):447–464.

Faruqui, Manaal and Padó, Sebastian. (2010). Training and evaluating a German named entity recognizer with semantic generalization. In *Proceedings of KONVENS 2010*, Saarbrücken, Germany.

Ferrucci, David and Lally, A. D. A. M. (2004). UIMA: an architectural approach to unstructured information processing in the corporate research environment. *Natural Language Engineering*, 10(3–4):327–348.

Hinrichs, Marie, Zastrow, Thomas, and Hinrichs, Erhard. (2010). Weblicht: Web-based lrt services in a distributed escience infrastructure. In *Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta.

Kerzel, Martina, Mittelbach, Jens, and Vitt, Thorsten. (2009). TextGrid: Virtuelle Arbeitsumgebung für die Geisteswissenschaften. *KI: Künstliche Intelligenz*, (4):36–39.

Manning, Christopher D., Raghavan, Prabhakar, and Schütze, Hinrich. (2008). *Introduction to information retrieval*. Cambridge University Press, New York.

Schmid, Helmut and Laws, Florian. (2008). Estimation of conditional probabilities with decision trees and an application to fine-grained POS tagging. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 777–784, Manchester, UK, August.

Schmid, Helmut. (1995). Improvements in part-of-speech tagging with an application to German. In *Proceedings of the ACL SIGDAT-Workshop*, pages 47–50.

Schmid, Helmut. (2009). Tokenizing and part-of-speech tagging. In Lüdeling, Anke, Kytö, Merja, and McEnery, Tony, editors, *Corpus Linguistics: An International Handbook*, Berlin. Walter de Gruyter.