4th Workshop on Challenges in the Management of Large Corpora

Workshop Programme

28 May 2016

14:00-16:00 – Session A

Introduction

Jochen Tiepmar,
CTS Text Miner – Text Mining Framework based on the Canonical Text Services Protocol

Jelke Bloem,
Evaluating Automatically Annotated Treebanks for Linguistic Research

Marcin Junczys-Dowmunt, Bruno Pouliquen and Christophe Mazenc,
COPPA V2.0: Corpus of Parallel Patent Applications. Building Large Parallel Corpora with GNU Make

Johannes Graën, Simon Clematide and Martin Volk,
Efficient Exploration of Translation Variants in Large Multiparallel Corpora using a Relational Database

16:00-16:30 Coffee break

16:30-18:00 – Session B

Adrien Barbaresi,
Collection and Indexation of Tweets with a Geographical Focus

Ruxandra Cosma, Dan Cristea, Marc Kupietz, Dan Tufiș and Andreas Witt,
DRuKoLA – Towards Contrastive German-Romanian Research based on Comparable Corpora

Svetla Koeva, Ivelina Stoyanova, Maria Todorova, Svetlozara Leseva and Tsvetana Dimitrova,
Metadata Extraction, Representation and Management within the Bulgarian National Corpus

Closing Remarks

Editors/Workshop Organizers

Piotr Bański, Marc Kupietz, Harald Lüngen, Andreas Witt	Institut für Deutsche Sprache, Mannheim
Adrien Barbaresi, Hanno Biber, Evelyn Breiteneder	Institute for Corpus Linguistics and Text Technology, Vienna
Simon Clematide	Institute of Computational Linguistics, Zurich

Workshop Programme Committee

Steve Cassidy	Macquarie University
Damir Ćavar	Indiana University, Bloomington
Isabella Chiari	Sapienza University of Rome
Dan Cristea	"Alexandru Ioan Cuza" University of Iaşi
Václav Cvrček	Charles University Prague
Koenraad De Smedt	University of Bergen
Tomaž Erjavec	Jožef Stefan Institute
Andrew Hardie	Lancaster University
Serge Heiden	ENS de Lyon
Nancy Ide	Vassar College
Miloš Jakubíček	Lexical Computing Ltd.
Piotr Pęzik	University of Łódź
Uwe Quasthoff	Leipzig University
Paul Rayson	Lancaster University
Laurent Romary	INRIA, DARIAH
Roland Schäfer	FU Berlin
Serge Sharoff	University of Leeds
Marko Tadić	University of Zagreb, Faculty of Humanities and Social Sciences
Ludovic Tanguy	University of Toulouse
Dan Tufiș	Romanian Academy, Bucharest
Tamás Váradi	Research Institute for Linguistics, Hungarian Academy of Sciences

Workshop Homepage

http://corpora.ids-mannheim.de/cmlc-2016.html

Table of contents

Author Index

Preface/Introduction

Creating very large corpora no longer appears to be a challenge. With the constantly growing amount of born-digital text – be it available on the web or only on the servers of publishing companies – and with the rising number of printed texts digitized by public institutions or technological giants such as Google, we may safely expect the upper limits of text collections to keep increasing for years to come. Although some of this was already true 20 years ago, we have a strong impression that the challenge has now shifted from an increase in terms of size to the effective and efficient processing of the large amounts of primary data and much larger amounts of annotation data.

On the one hand, some fundamental technical methods and strategies call for re-evaluation. These include, for example, efficient and sustainable curation of data, management of collections that span multiple volumes or that are distributed across several centres, innovative corpus architectures that maximize the usefulness of data, and techniques that allow for efficient search and analysis.

On the other hand, the new challenges require research into language-modelling methods and new corpus-linguistic methodologies that can make use of extremely large, structured datasets. These methodologies must re-address the tasks of investigating rare phenomena involving multiple lexical items, of finding and representing fine-grained sub-regularities, and of investigating variations within and across language domains. This should be accompanied by new methods to structure both content and search results, in order to, among others, cope with false positives, assess data quality, or ensure interoperability. Another much-needed research goal is visualization techniques that facilitate the interpretation of results and formulation of new hypotheses.

Due to the interest that the first meeting (at LREC 2012 in Istanbul) of CMLC enjoyed, the workshop has become a cyclic event. The second meeting took place at LREC again, in 2014 in Reykjavík; the third edition of CMLC was part of Corpus Linguistics 2015 in Lancaster. The upcoming fourth meeting will take place in Portorož, Slovenia, as part of LREC-2016.

CTS Text Miner
Text Mining Framework based on the Canonical Text Service Protocol

Jochen Tiepmar
ScaDS, Leipzig University
Ritterstrasse 9-13, 2.OG
04109 Leipzig
jtiepmar@informatik.uni-leipzig.de

Abstract

The purpose of this paper is to describe a modular framework for text mining that uses Canonical Text Service (CTS) as a data source. By combining standardized functionalities with standardized access to text data, this framework intends to reduce the heterogeneity of workflows in today's Digital Humanities and act as an important element of a text research infrastructure.
For this work the implementation of the CTS protocol described in (Tiepmar, 2015) is used. It uses advanced functionalities that are not part of the specifications of CTS. This means that, while most current modules should work with different implementations of the CTS protocol, it cannot be guaranteed that any future module will work.

Keywords: Text Mining, Infrastructure, Webservice

1 Introduction

One of the problems of text based Digital Humanities is its heterogeneity of data sources and methods. Data sources are made public in project specific ways and require the implementation of specific crawlers for each data set or a lot of manual work. Even though they are all modern projects, examples like Project Gutenberg[1], Perseus[2], Deutsches Text Archiv[3] and Eur Lex[4] each require individual ways to access the data. After the data is crawled, another problem occurs: the texts are not structured in a unified way. Each of the four examples uses a specific markup to structure their documents. DTA and Perseus offer texts in TEI/XML format, which is a text format that is often used to standardize a document's meta information. To access individual text units – for example lines – users still have to know in which way the structure is marked up in each document before being able to access it. Furthermore, if this information is not part of the TEI header, it is not possible to know, whether you should look for <l> or </lb> to access individual lines and <p> or for paragraphs. It may even be problematic to find out, how or if the document is structured in the first place.
If TEI/XML is not provided as text format, then it is already hard to divide the text from the document's meta information.
One possible solution is provided with the CTS proto-col by allowing generic access to documents and indi-rectly providing standardized access to the structure of documents. What is still missing is a collection of tools that use this access.

One of the significant features described in (Tiepmar, 2015) is CTS Cloning, which makes it possible to copy another CTS instance in parts or completely. If you can copy the text content and create a new instance of CTS, it is also possible to change it and convert the data into different formats. This paper describes the CTS Text Miner (CTS-TM), a framework with the goal of a selfsufficient, open and modular collection of methods that only require an instance of CTS as input.
Approaches to standardize text mining workflows al-ready exist and one may argue that this creates an ironic situation where a new standard is invented to solve the problem of too many standards. Yet, CTS, as it is specified in (Blackwell & Smith, 2014), was never developed as a text standard. The goal of CTS was to create a reference system based on the way that citation is done in common literature. This reference system is generic enough to be applicable to any text but on the other hand allows for exact citations as it is done for as long as literature is cited. Because of this strict and generic design, the protocol can be used as an access point for generic tools and therefore is considered as a good candidate for this work. This is also a fundamentally different approach than the ability to describe local documents in a meta format and use this meta description to convert full documents into compliant texts, as it is done for example by GATE[5].

[1] https://www.gutenberg.org/
[2] http://www.perseus.tufts.edu/hopper/
[3] http://deutschestextarchiv.de/
[4] http://eur-lex.europa.eu/homepage.html
[5] https://gate.ac.uk/

CTS requires the texts to be compliant with FRBR[6], which creates several significant limitations with re-gard to the texts that are compatible. This editorial limitation is ignored in this work that focuses on technical aspects.
Another benefit is that the URNs of CTS create a per-sistent ID system that connects text parts over multiple workflows making it possible to compare or complement results of one project with another. This also means that the ID that is returned as a result can be directly used to retrieve the corresponding text passage, which might for example be very useful for citation analysis. This persistent ID system combined with online availability can serve as a backbone for the interoperability as it is for example motivated in (Kalvesmaki, 2015).
In combination, CTS and CTS-TM make it possible to create an infrastructure where researchers can publish results and the corresponding data sets as instances of CTS and CTS-TM. Other researchers can easily reproduce the experiments by cloning the CTS instance and repeating the workflow with a given configuration of CTS-TM.

2 Requirements on CTS-TM

The target audience for this work consists of research-ers working with digital/algorithmic text analysis but includes researchers with little or no knowledge of or interest in the technical aspects of text analysis. This means that the actual work with the tools must be as easy and intuitive as possible. For these users the graphical user interface was implemented, enabling them to calculate results for the modules that are cur-rently implemented.
Since this framework is developed for a broad user base, it is hardly possible to include every important module for anyone. Especially the amount of para-metrization may vary widely or even contradict itself for different research groups. One may only need a basic default (working) parametrization for topic models while another may want to experiment with the parameters to find different results. This makes it impossible to develop individual workflows in a way that they appeal to anyone. What is possible is to design this framework in a way that users may develop their own modules or create better versions of existing ones. These modules can then become part of the main repository and this way be available for any other user as well. Creating individual modules or new workflows is something that ought to be done by users with more detailed technical knowledge, which is why this part is not fully fleshed out yet and will be completed in future work.
Language dependent algorithms often require trained models. Since CTS is not restricted to specific lan-guages, including such algorithms would require training data for any language. This means that for any of these modules and for any language that is supported, additional training files would have to be added to the package that may then not be required by individual users. To not overcomplicate things from the start, algorithms that rely on such additional resources are not added.
Performance is currently not optimized and optimiza-tion will be part of the future work.

3 Main Architecture

This chapter describes the current state of the main architecture of CTS-TM. Since this framework and the number and kind of modules will probably grow, some of the statements may not be true for future states of the implementation.
CTS-TM is available as a .war file that can be deployed by any compliant server, like Apache Tomcat. After deployment, a .jar executable can be found in the folder WEB-INF/lib and can be started using the command "java –jar CTSTM.jar". The configuration file conf.properties is stored next to this file and can be edited to (de)activate modules or configure properties of the input and individual workflows.
The first step of the workflow is to copy the data from the specified input. For this purpose URNs that are suitable for the given configuration are collected and stored locally with the corresponding text passage and meta information. It is advised for modules to use these local files to reduce the number of HTTP requests.
After the texts are stored locally, the individual mod-ules are started sequentially. Since parallelization might get included in this framework, only one module is running at the same time, even if it may be better to start several modules in parallel.
Many modules work with single tokens. At the mo-ment, tokenization is done by JAVAs default StringTokenizer, which works well for most cases. Support for additional tokenizers, for example Lucene's tokenizers, may be included in future work.
The data storage is not fixed and module implementers may use whatever they see fit and is compatible. In the current state, MySQL is used by most modules and most data is stored in one database. Additional modules may use this connection.
The results are available via HTTP requests. It is possible to request data while it is calculated but it is advised to wait until the import process is finished.
Indexing is done at the end of each module.

4 Modules

The following modules are currently included in CTS-TM. For each module, request and import functionalities are implemented.

4.1 Term_Document_Matrix

Term_Document_Matrix creates a MySQL table where tokens are listed and counted for any document. For example the entry

Docid	Termid	Token	Count
0	45	Mit	35

[6] http://www.ifla.org/publications/functional-requirements-for-bibliographic-records

describes a token “mit” which occurs 35 times in document 0.

4.2 Neighbors

The MySQL table of this module is filled similar to the table for Term_Document_Matrix, but instead of tokens, direct neighboring tokens are listed and counted. For example the entry

Docid	Id	Left	Right	Count
0	56	namhaften	Herrn	5

describes two tokens “namhaften” and “Herrn” that occur next to each other 5 times in document 0.

4.3 N_Grams

N-grams are sequences of tokens or characters with the length n. For example the sentence “The sun is shining” contains the 3-grams “The sun is” and “sun is shining”.
N is configurable and may contain multiple values that will result in a separate result set. For example the configuration “3&5” will result in one table for 3-grams and one table for 5-grams. The maximum length of one n-gram is 255, the key length of MySQL’s VARCHAR. This means that n * max_wordlength may not be longer than 255. One entry may look like

Docid	N_Gram	Count
1	aus_dem_hause	8

4.4 N_Gram_Reduce

This module creates a table for each n-gram table which contains the n-grams summed up for all docu-ments. Depending on the value for n, the number of entries is reduced significantly but the information about individual documents is reduced to the number of documents.

N_Gram	Count	DocCount
aus_dem_Hause	19	3

4.5 Term_Frequency

This module counts the occurrence of tokens in the dataset and the number of documents that contain this token. The following example describes the token “mit” that occurs 1108 times distributed over 10 doc-uments.

TokenId	Token	Count	DocCount
11994	Mit	1108	10

4.6 Term_Pruning & Document_Pruning

The purpose of these two modules is to use term fre-quency based pruning to reduce the number of tokens in several tables. Frequency based pruning is a com-mon method to reduce the number of tokens in a text corpus in Information Retrieval. The goal is to elimi-nate high frequency tokens for example to optimize text indices as it is done in (Carmel et al., 2001).
Document_Pruning only adds tokens that occur in less than a certain amount of documents. Term_Pruning adds tokens that are less frequent than a certain amount of the most frequent tokens.
For example, with the threshold 10 when using Document_Pruning, only tokens that occur in less than 90% of the documents are added. When using Term_Pruning, only terms that are less frequent than 10% of the most frequent tokens are added. The threshold for these methods can be configured.
This module requires Term_Frequency and Term_Document_Matrix as input. The results look similar to the results of Term_Document_Matrix.

Docid	Termid	Token	Count
0	106	Ewigkeit	2

4.7 Zipfian_Distribution

The module Zipfian_Distribution calculates the zipfian distribution as described in (Tullo & Hurford, 2003) for the full dataset. It requires Term_Frequency as input.

Rank	Term	Count
12	Wie	947

4.8 Stop_Words

Depending on the input that is available this module builds several tables with stop words, one for each Term_Pruning & Document_Pruning and one for the tokens of the Zipfian_Distribution with rank smaller than a configurable threshold. The stop word list based on the Zipfian_Distribution is also added as the default stopwords.txt file to be used by other modules.

Docid	Termid	Token	Count
0	45	Mit	35

Rank	Term	Count
12	Wie	947

4.9 Neighbor_Reduce

This module creates two tables similar to the table from Neighbors but without the entries which contain a left or right neighbor that is considered as a stop word by Term_Pruning & Document_Pruning.

4.10 Statistics

This table contains minimum, maximum, average, median and sum of the token count, tokens per document, document per token and terms per document.

4.11 Caching

For performance reasons some results are pre calculated and stored in local files. This includes the full token list or full zipfian distribution and other results that are expected to put a lot of load onto the database and are good candidates for data dumps.

4.12 Topic_Models

With the help of the JAVA library Mallet provided by (McCallum, 2002), topic models are calculated. The results are stored locally and in three MySQL tables and make it possible to request topics with their to-kens, topics with their documents and documents or tokens with their topics.

4.13 Document_Search_MySQL

This module stores the passages and their corresponding URNs and adds MySQL's fulltext index.

4.14 Document _Search_Lucene

This module builds a Lucene Index over all documents. The index is stored locally next to the file CTSTM.jar.

4.15 Doc_Search_Tokenlength_Signature

Instead of the text passage for the documents, this module creates a new passage that replaces the tokens with their lengths. For example the passage "The sun is shining." is indexed as "3 3 2 7 .".

4.16 Fulltext_Search

This module returns URLs for the exact passage that includes a given passage. These URLs combine the URL that is configured as input with the URN that is found. For it to work, one or many of the modules Document_Search_MySQL, Document_Search_Lucene, Doc_Search_Tokenlength_Signature or one of Term_Pruning & Document_Pruning are required as well as a CTS implementation that features fulltext search on text passage level. The required modules are used to find candidate documents. For each of these candidates, the text passage is searched by the CTS instance that is specified.
If multiple modules are configured as source for the document search, then candidate documents must be part of all their results sets.

4.17 Duplicate_Search

The most advanced module yet iterates through all the CTS URNs of every document in the CTS instance and uses their passages as input for Fulltext_Search. The goal is to create an undirected graph which connects duplicate or highly similar text passages between the input and the previously calculated data.

5 Requests

At the moment, the following requests are possible using HTTP communication. Optional [parameters] are added in round brackets.

5.1 Wordlists

- Stop word list based on [zipfian distribution | term pruning | document pruning]
- Stop words in [text] based on [zipfian distribution | term pruning | document pruning]
- [Text] minus the stop words based on [zipfian distribution | term pruning | document pruning]
- Zipfian distribution (from [rank1] to [rank2], with number of occurrences)
- Number of documents, types or tokens (for [token])
- List of documents, types or tokens (for [to-ken])
- Left or right neighbor token for [token] (plus number of occurrences)

5.2 N_Grams

- N_Grams for [n] (plus number of occurrences, up to [rank])
- documents containing [n_gram] for [n] (plus number of occurrences, up to [rank])

5.3 Search

- Documents or
- Text passages containing [text] using [mysql | lucene | term_pruning | doc_pruning | token-length_signature]

5.4 Topic Models

- Topics
- Topics plus tokens (from [rank1] to [rank2], with [weight])
- URNs for [topic] (with [weight])
- Tokens for [topic] (from [rank1] to [rank2], with [weight])
- Topics for [urn] (with [weight])

6 Evaluation

Three data sets are used for this evaluation:
1) The TED Subtitle Corpus (TED) that was published as a CTS instance in 2015 contains 52'987 relatively small documents in 105 languages. For the benchmark, only English documents were included resulting in 1938 document with 4'172'395 tokens / 56'742 types.
2) A snapshot from the text corpus "Deutsches Text Archiv" (DTA) from November 2014 as it was pub-lished as an instance of CTS. Only the normalized documents are used resulting in 1712 documents with 114'711'190 tokens / 1'565'612 types.
3) The Parallel Bible Corpus (PBC) as it was published as a CTS instance in 2015.
The test system is a virtual machine with 4 GB of memory, 6 GB Swap, a Quad-Core AMD Opteron(tm) Processor 8356 and a 350 GB ATA disk. Every request was sent via localhost and the system was rebooted before the benchmarks were started. SQL was provided by MariaDB 5.5.47 (Ubuntu 14.04.01). Apache 7.0.28 was used as the server with JAVA 1.7.0-55-b14. The default configuration was not changed. For better readability numbers are rounded to integers.

6.1 Performance

To evaluate performance, two instances of CTS-TM were created – one based on the data from the DTA CTS filtered by ".norm:" and one based on the TED CTS filtered by ".en:". For each of these instances the tokens, types, zipfian distribution and zipfian distribution from rank 2 to rank 100 were requested globally and for each document.
Table 1 and Table 2 show the number of elements and the response times for every global request. The global token list for DTA resulted in a memory error which would have to be fixed in the server configuration. The correct value is 114'711'190.

	Tokens	types	zipf	zipf2
TED	4’172’395	56’742	56’742	99
DTA	N/A	1’565’612	1’565’612	99

Table 1 Number of elements in result set

	Tokens	types	zipf	zipf2
TED	7’565	749	241	13
DTA	1’860	1’011’069	4’631	86

Table 2 Response time in MS

Table 3 and Table 4 show minimum, average and maximum number of elements and response times for every request for every document in TED instance.

	Tokens	types	zipf	zipf2
min	2	2	2	1
avg	2’153	612	612	98
max	6’618	1’362	1’362	99

Table 3 P: TED: number of elements in result set per URN

	Tokens	types	zipf	zipf2
min	11	9	9	8
avg	18	13	14	12
max	120	127	18	44

Table 4 P: TED: response time per URN in MS

Table 5 and Table 6 show results for the same bench-mark using the DTA data.

	tokens	types	zipf	zipf2
min	73	60	60	59
avg	67’004	8’228	8’227	98
max	1’082’829	51’430	51’429	99

Table 5 P: DTA: number of elements in result set per URN

	tokens	types	zipf	zipf2
min	8	8	8	8
avg	38	33	59	37
max	1865	311	845	444

Table 6 P: DTA: response time per URN in MS

The different values for zipf and types in
Table 6 might indicate an encoding related bug and require further investigation.

These results show that the performance of the system is good enough to be used productively and scales well. Both instances of CTS-TM achieve an average response time that is well below 100 MS. The impact of the higher number of tokens / types per document in DTA is not very big but measureable. DTA‘s global token list could not be requested because the string was too big to be handled by the server given the default configuration. The response times for global results will be optimized with locally stored pre calculated files (caches).

6.2 Document Search

CTS-TM provides two kinds of text search methods: document search and text passage search. Because the text passage search uses an external resource, it is not evaluated in this work. The results of the evaluation are also important for the text passage search because the document search influences the number of candidate documents that the CTS instance has to consider. Lower response times and smaller result sets have positive effects on the response times of the text passage search.

The score that is required to be considered as a candidate by Lucene is 0.1.

Evaluation was done similar to 6.1. The functions that were used are separated by the search methods that are available: lucene, term_pruning, doc_pruning and tokenlength_signature. MySQL could not be evaluated because the SQL Fulltext index was not supported by the test system. The first CTS text part of the original document was used as the query text.

	signature	termprun	docprun	lucene
min	1	1	1	0
avg	161	353	339	134
max	1938	1923	1923	1000

Table 7 TS: TED: number of elements in result set per URN

	signature	termprun	docprun	Lucene
min	8	10	9	2
avg	52	353	339	103
max	192	874	631	219

Table 8 TS: TED: response time per URN in MS

Table 9 and Table 10 show the benchmark results for the DTA data.

	signature	termprun	docprun	lucene
min	0	0	0	0
avg	340	50	50	39
max	1712	1672	1672	1000

Table 9 TS: DTA: number of elements in result set per URN

	Signature	termprun	docprun	Lucene
min	7	9	9	3
avg	720	807	769	1851
max	4957	31342	10340	21678

Table 10 TS: DTA: response time per URN in MS

Document search performs relatively well for any of the methods. That the response time for the token length signature is faster than the response time for the method using the Lucene index is surprising as are Lucene’s relatively bad response times for the DTA data set. Lucene managed to create the smallest candidate lists. However, in 253 TED requests, it did not return any result while any other method did find something.

Table 11 shows the amount of empty results for DTA.

signature	termprun	docprun	Lucene
23	1	1	412

Table 11 Number of empty results

Since every text passage was definitely part of the data set, these empty results might indicate errors.
This does not mean that Lucene performs generally worse than the other solutions, only as it is implemented in CTS-TM.

6.3 Research Value

Using Duplicate_Search a text re-use analysis was done that compared the German translations of the bible against the CTS-TM calculated with the DTA CTS with the goal to find bible citations in DTA. For this purpose, documents were filtered using the token length signature described in 4.15.
Table 12 shows the number of cited passages from PBC and citations in DTA for each of the five German bible translations that are part of PBC.

	1	2	3	4	5
PBC	32	27	361	271	57
DTA	5954	272	1667	1107	479

Table 12 Duplicate text passages between PBC & DTA
1 = elberfelder1871 3 = luther1545
2 = elberfelder1905 4 = luther1912
5 = schlachter

As it was expected, citations for bible translations by Luther were most prominent. The high number of citations of elberfelder1871 is the result of passages like “und 61000 Esel” or “und 72000 Rinder”. Numbers are deleted and “und” is considered a stop word. This results in passages like “Rinder” and “Esel”, which occur often. In elberfelder1905, these numbers are spelled in full[7].
Because of the connection to CTS, each of the citations is referenced as a CTS URN and can be used to retrieve the corresponding text passage. A simple visualization based on a prototype of (Reckziegel et al., 2016)’s CTRaCE uses this connection to list each citation with links to the corresponding text passage in DTA:

Am Anfang schuf Gott Himmel und Erde .
urn:cts:pbc:deu.luther1545:1.1.1
-- am anfang schuf gott himmel und erde
-- im anfang schuf gott himmel und erde[8]

Furthermore, since PBC is a parallel corpus, the URNs can be used to align the citations over different translations in one language, for example resulting from different use of metaphors. The text passage "ich wache, und bin wie ein einsamer Vogel auf dem Dache." from elberfelder1871 was found as a citation in DTA. The corresponding text passage "Ich bin gleich wie eine Rohrdrommel in der Wüste; ich bin gleich wie ein Käuzlein in den verstörten Stätten." in luther1545 was not found but can be associated when the results for the different translations are aligned using the alignment of CTS URNs. Furthermore, this means that any translation of the bible that is part of PBC can potentially be aligned against these results, making it possible to create citation graphs based on DTA for each of the 831 translations. And finally, because every result that is calculated with CTS-TM shares the same CTS URNs as identifiers, these results can be enriched with any other result of CTS-TM or tool that is developed with compatibility for the CTS protocol.
The results show high precision but low recall. High precision can be implied because each citation is an exact reference to this text passage in DTA if stop words and numbers are ignored. Recall is hard to measure because there does not exist a complete list of bible citations in DTA. DTA includes many phrases with variations of single tokens. For example, the phrase "Am Anfang schuf Gott Himmel und Erde." also occurs as "Am Anfange schuf Gott Himmel und Erde." or "Am Anfange schuff Gott Himmel und Erde.". These variations are not included in the results but can be covered by including editing distances in the fulltext search.
Another issue is based on the format of PBC. Duplicate_Search uses the smallest referencable CTS text parts as input. These may be too big for certain famous text passages like "Nehmet, esset, dies ist mein Leib", which appears nowhere in the results. However, when searched explicitly, this phrase is found 932 times in DTA. The problem is that this phrase is always included in bigger contexts like "Da sie aber aßen , nahm Jesus das Brot , dankete und brach's und gab's den Jüngern und sprach : Nehmet , esset ; das ist mein Leib .", which do not appear in DTA. Segmentation techniques can be applied to divide phrases into smaller text parts but this would require additional language dependent resources. The easiest way to include such phrases is to create an additional edition with this in mind and use it as input.
The resulting citation graph is not directed, which is not a problem in this case. It is unlikely that the bible re-used a passage from DTA. For future workflows the publication dates that are available as meta information in CTS can be used to create directed graphs.
Much more focused work concerning text re-use was done by (Büchler, 2013) or can be done with the help of Winnowing described in (Schleimer & Wilkerson & Aiken., 2003). The goal of this evaulation was to illustrate the benefit of the interoperability that is provided by the shared set of identifiers by combining an algorithmic workflow of text re-use with an algorithmic workflow of text alignment and a generic visualization tool.
As proof of concept, the binaries, source code and configuration for CTS-TM, as it was evaluated in this work, are available online[9] so that anyone can repeat the evaluation locally.

[7] See urn:cts:pbc:deu.elberfelder1905:4.31.34 and urn:cts:pbc:deu.elberfelder1905:4.31.33

[8] Complete results for luther1545: http://ctstm.informatik.uni-leipzig.de:8080/tr_dh/

[9] http://ctstest.informatik.uni-leipzig.de/eval/ctstm_release.zip

7 Conclusion

As it is clearly shown in 6.3, the connection of CTS with standardized workflows shows a lot of potential.

Especially the combination of separate results and tools by the usage of a shared set of identifiers and the fact that all of the results can be easily recreated using data that is publicly available, creates a transparent and interoperable environment. When working with parallel text corpora, the research results can even be shared across language barriers.

Future Work might include the implementation of additional or alternative modules and the addition of generic visualization modules, for example for bags of words, links between documents or topic models. Additional tokenizers can be implemented to enhance import functionalities. Since Lucene is already part of this framework, implementing compatibility with Lucene's tokenizers might be the easiest way to do this while making sure that future state of the art tokenizers can be included without much implementation effort.

Additionally, CTS-TM must be further connected to established text workflow frameworks and document databases like Sketch Engine, GATE and other text mining oriented data storages as it was already done with Lucene and MySQL's fulltext index. The connection to workflow orientated tools like KNIME might also create a lot of opportunities for interoperability between different tools.

8 Acknowledgements

This work was funded by the
German Federal Ministry of Education and Research
within the project
Competence Center for Scalable Data Services and Solutions (ScaDS)
Dresden/Leipzig (BMBF 01IS14014B)

9 Bibliographical References

Blackwell, C. & Smith, N. (2014). Canonical Text Services protocol specification. Retrieved from http://folio.furman.edu/projects/citedocs/ctsurn/ and http://folio.furman.edu/projects/citedocs/cts/ 2015, February 19.

Büchler, M. (2013). Informationstechnische Aspekte des Historical Text Re-use (English: Computational Aspects of Historical Text Re-use). PhD Thesis. Leipzig University.

Carmel D. & Cohen D. & Fagin R. & Farchi E. & Herscovici M. & Maarek Y. & Soffer A. (2001). Static Index Pruning for Information Retrieval Systems. In Proc. ACM SIGIR.

IFLA Study Group on the Functional Requirements for Bibliographic Records (1998). Functional Requirements on bibliographic records: final report. In UBCIM publications; new series, vol. 19. München, K.G. Saur.

Kalvesmaki J. (2015). Three Ways to Enhance the Interoperability of Cross-References in TEI XML. In Proceedings of the Symposium on Cultural Heritage Markup. Balisage Series on Markup Technologies, vol. 16.

McCallum, A. (2002) MALLET: A Machine Learning for Language Toolkit. http://mallet.cs.umass.edu. 2002.

Reckziegel, M., Jänicke S., Scheuermann G. (2016). CTRaCE: Canonical Text Reader and Citation Exporter. To appear in Proceedings of the Digital Humanities, Krakow.

Schleimer S. & Wilkerson D. & Aiken A.(2003). Winowing: local algorithms for document fingerprinting. In Proc. Of the 2003 ACM SIGMOD Intl. Conf. on Management of data, pages 76-85.

Tiepmar J. (2015) Release of the MySQL based implementation of the CTS protocol. In Proceedings of the 3rd Workshop on Challenges in the Management of Large Corpora (CMLC-3).

Tullo C & Hurford J. (2003). Modelling Zipfian Distribution in Language. In Kirby, S. Language Evolution and Computation, Proceedings of the workshop at ESSLLI.

Evaluating Automatically Annotated Treebanks for Linguistic Research

Jelke Bloem
University of Amsterdam
1012 VB Amsterdam, Netherlands
j.bloem@uva.nl

Abstract

This study discusses evaluation methods for linguists to use when employing an automatically annotated treebank as a source of linguistic evidence. While treebanks are usually evaluated with a general measure over all the data, linguistic studies often focus on a particular construction or a group of structures. To judge the quality of linguistic evidence in this case, it would be beneficial to estimate annotation quality over all instances of a particular construction. I discuss the relative advantages and disadvantages of four approaches to this type of evaluation: manual evaluation of the results, manual evaluation of the text, falling back to simpler annotation and searching for particular instances of the construction. Furthermore, I illustrate the approaches using an example from Dutch linguistics, two-verb cluster constructions, and estimate precision and recall for this construction on a large automatically annotated treebank of Dutch. From this, I conclude that a combination of approaches on samples from the treebank can be used to estimate the accuracy of the annotation for the construction of interest. This allows researchers to make more definite linguistic claims on the basis of data from automatically annotated treebanks.

Keywords: Evaluation, Linguistics, Automatic Annotation

1. Introduction

This paper addresses an issue that is important for the application of large, automatically annotated corpora to linguistic research. A disadvantage of corpus-based methods in linguistics is that many phenomena of interest to theoretical linguists are used infrequently in naturalistic speech, and therefore are less likely to occur in smaller corpora. Automatically annotated linguistic resources contain the 'big data' that is necessary to study these phenomena empirically, but they inevitably contain errors as well, due to the imperfect natural language processing tools that were used to annotate them. As linguists are increasingly using large corpora as a source of empirical evidence, these issues have been acknowledged, but not explored systematically. In this work, I discuss four different approaches to evaluating data from automatically parsed corpora when a particular linguistic phenomenon is being studied. Current methods for evaluating the quality of annotation are too general for the purposes of studying a particular construction, and only measure the overall accuracy of the annotation of a corpus. While I use treebanks to illustrate the approaches because they seem to be the most common type of automatically annotated language resource, the approaches are also applicable to language resources created using other forms of automatic annotation, such as part-of-speech tagged historical texts or semantically parsed corpora.

1.1. Automatically Annotated Treebanks

Treebanks are text corpora that have been enriched with syntax trees or syntactic graphs (e.g. when dependency grammars are used), allowing linguists to search those texts for particular syntactic constructions and morphological features. Such queries will result in a list of only those constructions, which is much easier to study than an entire text. Due to advances in natural language processing, it has become possible to syntactically parse large amounts of text automatically, with fairly good accuracy. This has resulted in the creation of treebanks that are much larger than traditional, manually annotated corpora. From a linguistic perspective, the main advantage of these large-scale treebanks over manually annotated corpora is that they can be used to investigate rare constructions, co-occurrence patterns of uncommon words or small probabilistic effects. They also provide larger sample sizes or lists of examples of naturalistic data for more common linguistic phenomena.

Probabilistic effects in language have been discussed particularly in the study of alternations, i.e. multiple near-synonymous constructions that form two grammatical options for expressing the same meaning. Corpus studies of such phenomena have revealed that a number of factors from various domains of language (i.e. phonetics, semantics, pragmatics) may affect the choice between alternative constructions in such an alternation to varying degrees. The size of these effects can be interpreted as probabilities. The linguistic implications of the observation of such effects have been discussed by Bresnan (2007), who studied the English dative alternation. This multifactorial study experimentally tested whether probabilistic effects found in a previous corpus study corresponded to speaker preferences in a rating experiment between the two constructions of the dative alternation, i.e. 'I gave her the book' and 'I gave the book to her'. An earlier example of a multifactorial study of a linguistic optionality can be found in Gries (2001), who shows that many factors affect particle placement in English. In this study, Gries also computes effect sizes to quantify how influential the factors are, though the term 'probability' is not explicitly mentioned.

The main disadvantage of automatically annotated corpora is the error rate. While manually annotated or manually checked treebanks contain some errors, automatic annotation comes at the cost of annotation accuracy. The errors made by the parser will include systematic errors, where the parser has more difficulty with certain types of constructions than others. The parser may even be unable to annotate a particular construction correctly, thereby failing to provide the necessary means to search for it in a large corpus. Therefore,

when using automatically annotated treebanks for linguistic study, some sort of evaluation is necessary to make sure that the construction of interest was annotated correctly, or at least well enough for the purposes of the study. While the accuracy rate of the parser that was used to annotate the treebank is usually known, such a general measure of evaluation is not meaningful for most linguistic studies.

1.2. Construction-specific Querying

When studying a particular linguistic phenomenon or construction in a corpus, it may be more relevant to view the task as a form of information retrieval — all of the sentences instantiating the construction have to be retrieved from a larger data source (the corpus). The main difference with most information retrieval tasks is that the success of the search process depends on the quality of the annotation, rather than the quality of the search algorithm or the query. Nevertheless, I will use two basic measures from information retrieval, precision and recall, to illustrate the various approaches to evaluating the annotation quality of a corpus for a particular construction. In the context of this paper, **precision** is defined as the fraction of results from a corpus query that are instances of the construction that is being searched for, while **recall** is defined as the fraction of instances of the construction in the corpus that are retrieved. I will assume that corpus queries are perfectly written following the annotation format of the corpus being queried to specify exactly what the researcher wants to retrieve. Under this assumption, any imperfections in precision and recall occur only due to incorrect annotation. In reality, other issues may affect precision and recall as well, such as inaccurate formulation of the query or a lack of distinction between certain phenomena in the annotation format of the corpus. Since those would be problems of information retrieval rather than automatic annotation quality, I will not focus on them in this discussion.

2. Linguistic Studies Using Automatically Annotated Treebanks

Automatically annotated treebanks are a useful source of information in any study where large sample sizes are beneficial. Such treebanks have been made available for various languages. For Dutch, there is the 700 million word Lassy Large treebank (van Noord et al., 2013). For German, the 200 million word TüPP-D/Z (Müller, 2004) is available with automatic annotation. For English, the Google Books n-gram corpus (Lin et al., 2012) has been annotated syntactically, as well as the 4 billion word Gigaword v5 corpus (Napoles et al., 2012), to name but a few. Treebanks for a specific domain or language can be created as long as an automatic parser is available. This is the case for many major languages. Efforts have been made to make this technology more accessible to linguists who do not necessarily have a technical background, using techniques like example-based querying for treebanks (Augustinus et al., 2012), or systems where researchers can upload their own corpora to be automatically annotated, such as PaQu (Odijk, 2015).

These treebanks have already been used to study various linguistic phenomena. For Dutch, several applications of the Lassy Large treebank are discussed by van Noord and Bouma (2009). A study of extraposition of comparative objects by van der Beek et al. (2002) was used to illustrate the grammar used by the Alpino parser, but it attracted criticism for allowing too much extraposition. As evidenced by a note (van der Beek et al., 2002, 364, note 8), a reviewer claimed that such extraposition was not possible from the front of the sentence (the topic), however, a search of the large corpus revealed that such sentences were in fact being used in particular contexts. It was judged to be a probabilistic phenomenon, more or less acceptable depending on various factors. This shows that automatically annotated treebanks can also be used to refute claims based on linguistic theory. Bastiaanse and Bouma (2007) used syntactic structures from the treebank to argue that patients with Broca's aphasia have difficulty with constructions of higher linguistic complexity, rather than due to a frequency phenomenon. Bouma and Spenader (2008) studied the distribution of the Dutch reflexives *zich* and *zichzelf* with regards to the verbs with which they co-occur, where different verbs can select one or both of the options. These are examples of studying rare constructions or co-occurrence patterns, a task that automatically annotated treebanks are particularly suitable for.

Another such task is the study of probabilistic effects. Bloem et al. (2014) used a part of the Lassy Large treebank to study Dutch verb cluster constructions, a word order variation in which a variety of factors play a probabilistic role. Like English, the Dutch language may use auxiliary verbs to express features such as tense and aspect. In verb-final subordinate clauses, these verbs are grouped together at the end of the clause, and in main clauses, the first (finite) verb goes to the second position while the others are grouped together at the end of the clause. Interestingly, in subordinate clause two-verb clusters, both logical orders are grammatical:

(1) Ik zei dat ik het **gehoord heb**
I said that I it heard have
'I said that I have heard it.'

(2) Ik zei dat ik het **heb gehoord**
I said that I it have heard
'I said that I have heard it.'

Speakers may choose between the orders depending on a variety of factors relating to discourse, semantics, mode of communication or processing complexity (De Sutter, 2009; Bloem et al., 2014). Larger clusters of verbs are also possible, but not all of the logical orders are grammatical when three or more verbs are involved, although there is still variation.

Studying this phenomenon involves searching the treebank for groups of verbs in a particular syntactic configuration: an auxiliary or modal verb heading a participial or infinitival main verb. This study also replicates earlier work on a manually annotated corpus (De Sutter, 2009), showing that the errors caused by the automatic parsing are not necessarily a problem for linguistic study, although a few factors (i.e. word stress patterns) could not be studied due to the nature of the annotation that can be found in a treebank of written texts. While the manual study involved 2.390 instances of verb clusters, a sample of 411.623 clusters was gathered from the treebank.

Odijk (2015) showed that automatic annotation can even be used to study child utterances from the Dutch CHILDES corpus. This corpus was parsed with the Alpino parser for Dutch, even though this parser has not been trained on child language data. Spoken child language is a rather different domain than adult written language, making parsing errors likely. Nevertheless, a study of three near-synonymous Dutch degree modifiers that translate to 'very' was conducted on this data, along with an evaluation. The interesting thing about these modifiers is that two of them, *erg* and *zeer*, are used with adjectival, verbal and adpositional predicates, while one, *heel*, is only used with adjectival predicates. It is not clear how children acquire this difference. A corpus of child utterances and child-directed speech with syntactic information may reveal how much evidence there is for these constructions in child language. Useful results were obtained despite the issues, likely due to the focus on a particular linguistic phenomenon involving modifiers rather than larger syntactic structures — the two most common of the three degree modifiers were found with high accuracy.

Using the TüPP-D/Z treebank, auxiliary fronting in German three-verb clusters was studied (Hinrichs and Beck, 2013). Since three-verb clusters in subordinate clauses are a somewhat rare construction, and auxiliary fronting inside of them even more so, the massive size of the corpus was a requirement to be able to find enough instances. The authors observe what verbs participate in the construction and compare the treebank data to (much more sparse) information from diachronic corpora. For English, Lehmann and Schneider (2012) used a 580 million word dependency-parsed corpus to study the influence of specific lexical types on the English dative alternation. These types consist of 'triplets' of words: a ditransitive verb, a direct object head and an indirect object head — these slots are all filled with open-class words, requiring massive amounts of data to study.

3. Current Approaches to Evaluation

The quality of automatically annotated treebanks is usually evaluated by testing the performance of the parser that was used to create it. Therefore, treebanks are evaluated in the same way as parsers, using an overall accuracy score such as the word-based Attachment Score. This is the percentage of words that have been assigned the correct head in the syntactic structure (sentence-based variations or variations that include dependency labeling also exist). The Alpino parser (van Noord et al., 2006) that was used to create the Dutch Lassy Large corpus was evaluated using Concept Accuracy (the proportion of correct labeled dependencies) as a measure. A part of the corpus containing texts from various domains (e.g. books, newspaper texts) was manually verified in order to have a gold standard to compare against. This resulted in an accuracy score of 86.52%, but with clear variation across different domains (van Noord, 2009). Studies based on the corpus often report this score as a measure of quality.

However, even this is too general for the purposes of linguistic research. Rather than some domain of text, a researcher is primarily interested in one particular construction, and wants to know how accurately that particular construction was parsed in the corpus. If the parser often errs in labeling adjectives, this does not matter if one wants to investigate reflexives, but it would be a major problem for a study of adjectives. Parser errors cannot be entirely dismissed as random variation, some of the errors are likely to be systematic due to the nature of (most) syntactic parsing as a probabilistic task based on statistical learning.

One obvious consequence of this is that a parser is more likely to make mistakes when parsing rare phenomena, for which there was little evidence in the parser's training data. Phenomena that are of interest to linguists are often rare. Related to this is the fact that parsers generally perform worse on longer sentences, as shown in van Noord et al. (2006, 11, Fig. 5) for the Dutch Alpino parser. Sentence length is sometimes considered as a probabilistic processing effect in multifactorial linguistic studies, so this should also be considered. More errors occur when there is more ambiguity, regardless of whether the ambiguity is caused by semantic or syntactic factors. Multi-word units (idiomatic expressions) are also known to cause parsing errors (Nivre and Nilsson, 2004), but on the other hand, a parser may have been specifically improved to deal with multi-word units. Text types that are different than what the parser was trained on, such as the child utterances in the study by Odijk (2015), may also cause a higher error rate. Lastly, when the original text contains errors or unusual spelling, a parser is also likely to make more annotation errors.

Due to this possibility of systematic errors which may introduce more errors for certain constructions than for others, I believe that semi-automatic or manual construction-specific evaluation is necessary, using the knowledge of linguistic experts. Such an evaluation will provide insight into the quality of data gathered from automatically annotated treebanks for the purpose of linguistic study.

Some studies using automatically annotated treebanks have taken this approach. For example, Odijk (2015), in his CHILDES study, compares the parser accuracy for the specific words being studied against a manually annotated gold standard. However, such a gold standard is not always available, and the manual annotation was also found to contain errors. These errors were found by looking up the words manually, which is not possible if one is investigating more general constructions that can involve many words types, instead of particular words. Furthermore, the data set of child utterances of the constructions being investigated was fairly small, making a thorough manual evaluation more feasible than on large automatically annotated corpora. Therefore, this approach to evaluation is not always applicable. Bloem et al. (2014) took a semi-automatic approach by manually verifying a portion of the results of their syntactic queries for verb clusters. While the precision of the results can be measured with such an evaluation, it does not address the issue of recall. Any relevant construction that was annotated incorrectly and therefore missed by the querying procedure, will not be in the sample. In other studies, i.e. Hinrichs and Beck (2013), the paper does not address the issue of construction-specific evaluation at all.

In the next section I will discuss four possible approaches to construction-specific evaluation for linguistic research using an automatically annotated corpus. I will illustrate the four approaches with examples from the Dutch verb cluster

research described by Bloem et al. (2014), who conducted their research on the automatically annotated Lassy Large corpus. In listing these approaches. I am assuming that the linguist is faced with a corpus that is the end product of automatic annotation. This hypothetical researcher does not have access to, or is not able to use the tools that were used to annotate it, i.e. the methods do not require much technical knowledge. Without this restriction, other approaches could be taken, and have already been taken, such as re-training and/or evaluating the parser on an adapted text, using or creating a different annotation tool that is designed to target the construction of interest specifically, or simply parsing a large number of instances of the construction being studied and evaluating the parser's performance on that procedure. However, it is unlikely that someone whose main interest is linguistic research would have the knowledge or motivation to perform such procedures.

4. Linguistically Informed Evaluation

The main difficulty of this task, evaluating some subset of a large corpus (i.e. all verb clusters), is in gaining insight into precision and recall at the same time. The four approaches discussed here have various strengths and weaknesses relating to these two measures that I will discuss. An overview of the four approaches discussed in this section and their relative benefits is shown in table 1.

4.1. Manual Evaluation of the Results

The most obvious approach is a complete manual evaluation of the results by a linguist. This involves first formulating a query that matches a specific construction, and manually inspecting the results of the search. Any result that matched the query but was not actually an instance of the construction being studied, whether it was due to an annotation error or an imprecise query, is marked as false, and others as correct. A percentage can then be calculated, which represents the precision score of the query. However, this method has several disadvantages. Firstly, it may take a lot of time and resources to evaluate all results extracted from a large corpus in this way — Bloem et al. (2014) automatically extract 411.623 verb clusters from the 145 million word Wikipedia part of the Lassy Large corpus, too many to verify manually in any reasonable time frame. A representative sample of the results would have to be used. Secondly, this method may still miss constructions that were systematically misparsed. For example, if a researcher is searching for verb clusters but verbs in a particular type of cluster have been mistagged as adjectives, a search query for verb clusters will not find those mistagged instances, and the researcher will not know of their existence. The precision of the results can be measured with such an evaluation, but not the recall.

I have tested this method on the first 10.000 sentences of the Wikipedia section of the Lassy Large Large treebank using two-verb auxiliary clusters from subordinate clauses, as shown in (1) and (2) as an example construction. This sample of the corpus contains 193.378 tokens, covering 0.13% of the Wikipedia section of Lassy Large. A syntactic search for the target construction yielded 315 matching verb clusters, of which five were found to arguably constitute errors — these five verb clusters all had adjectival instead of verbal participles. An example of that would be 'He thought the door was closed', where 'closed' can be an adjective as well as a verb, and these five cases were annotated as adjectives. However, the fact that they could be verbs was also available in the annotation, so these five examples may not be errors depending on your theoretical perspective. Therefore, the precision of the annotation for this part of the corpus is $\frac{310}{315} = 0.984$, or 1. From this we can conclude that two-verb clusters were likely parsed with very high precision by the Alpino parser when this corpus was created.

4.2. Manual Evaluation of the Text

To solve the recall problem, it may be possible to do a manual evaluation of the text. By reading the original corpus text rather than just the results of a search query, even instances of the construction of interest that are completely misparsed can be found by the linguist. However, this is extremely labor-intensive — one will have to read a lot of text to find just one instance of a rare construction, even if only a part of the corpus is evaluated in this way. This takes away the main advantage of using a large automatically parsed treebank, and even if only a representative sample of the corpus is read, this is only feasible for common constructions. Therefore, I have chosen not to demonstrate this approach.

4.3. Fall Back to Simpler Annotation

Another solution is to fall back to a simpler annotation layer. It is generally the case that annotation of larger structures is more difficult. Lemmatizing and tagging (assigning a word class) only involves words, while parsing adds syntactic structure over multiple words. Queries based exclusively on word class will therefore result in fewer errors than searching on the basis of syntactic structure. For example, to retrieve verb cluster construction one would normally want to find a verb that is the head of another verb. But in this way, verbs that were attached incorrectly will erroneously be skipped. If the researcher simply searches for two verbs positioned next to each other in the linear order, these skipped verbs would also be included, at the cost of retrieving verbs that are next to each other coincidentally (i.e. as part of two different clauses) or as part of a larger structure. Comparing the result of the two procedures will produce a list of 'suspicious' instances, which can be evaluated manually (to be either included or excluded from the study) with less effort and better recall than when the results of a regular corpus query are manually evaluated. This does mean that the linguist will have to come up with some sort of word-class-based approximation of the construction under study using their knowledge of the language.

This approach is somewhat comparable to what the Sketch Engine does, as introduced by Kilgarriff et al. (2004). The Sketch Engine is a tool aimed specifically at lexicographers. It can extract collocation information and other information that is interesting for lexicography from a corpus, while ignoring other aspects of the annotation. It has been applied to a variety of corpora, including automatically annotated ones. However, it does make use of some syntactic structure annotation (which is not simple), namely to identify grammatical relations of collocations.

I have again tested this method on the first 10.000 sentences

Approach	Weaknesses	Strengths
Manual evaluation of the results	No recall, somewhat costly	Precision measure
Manual evaluation of the text	Extremely costly	Precision & recall
Fall back to simpler annotation	Misses POS-tagging errors	Recall measure
Search for particular instances	Hard to generalize result	Recall measure

Table 1: Overview of the strengths and weaknesses of each approach.

Error category	Frequency	Percentage
Part of longer cluster	56	74.67%
Parsing error	7	9.33%
Query error	12	16.00%
Total differences	75	100%

Table 2: Results of a comparison between syntactic search and POS-based search, listing the verb clusters found only by the latter one.

of the Wikipedia section of Lassy Large, comparing the result of a syntactic search with that of a part-of-speech (POS) based search using only the features of word class, lexical category and linear position in the sentence. The results of this are shown in table 2. I identified all results that were retrieved by the POS-based search but not by the syntactic search, and manually verified them. There were 75 such results in total. In 56 cases, the query had actually matched a group of two verbs that was part of a larger verb cluster of more than two verbs. Since only two-verb clusters, not three or four verb clusters are the target construction, these are not errors. It is difficult to avoid getting results from larger clusters when using POS-based search, since the difference is syntactic. In seven cases, there was an actual two-verb cluster that had been misparsed. These cases were mostly located in very long sentences with many parsing errors. The syntactic search had missed these, indicating a recall issue. A further 12 results also contained actual two-verb clusters and were annotated correctly, but were not identified by the syntactic search. This indicates a problem with the syntactic query rather than with the annotation. They can be considered retrieval errors, not annotation errors. Most of them involved verbal particles directly before or after the verb cluster, which I did not consider when formulating the syntactic query. Detecting such errors using this method can help the researcher to refine their queries. Overall, to the 315 verb clusters that were found in the previous section, we can now add $7 + 12$ new ones that were not identified by the syntactic search. This also allows me to compute the recall over this part of the corpus: $\frac{315}{334} = 0.943$, or $\frac{315}{322} = 0.978$ if we do not wish to consider the query errors as a recall problem. Again, this is only an estimation of recall, calculated over a fraction of the entire corpus and using manual comparison of the results. Furthermore, this estimate does not take into account that the part-of-speech tagging may also contain errors. Automatically annotated part-of-speech tags are not perfect either, they are just more correct than the parse trees. The final approach I will discuss does not make this assumption, but instead circumvents the annotation as much as possible.

4.4. Search for Particular Instances

It is possible to search for particular instances (types) of the construction without relying on the annotation at all. The linguist can choose some representative instances of the construction and search for it directly. For example, when searching for Dutch verb clusters, one could simply search the corpus for the string *hebben gehad* 'have had', one of many possible combinations of verbs. I will call this a 'string query', as opposed to a 'syntactic query' that one would normally perform on a treebank. This will only result in a limited number of results, but in a large automatically annotated corpus there can still be many results for a specific combination of words, even if the total number of instances of the general construction (verb-verb combinations in this case) is much larger. This string query does not rely on any syntactic annotation, and it would therefore find the verbs even if they were annotated completely erroneously, i.e. as a preposition heading a preposition. These results for the string *hebben gehad* can then be compared to results for the syntactic structure of 'hebben gehad' with these particular words to see whether there is a recall problem: if an example occurs in the string query but not in the syntactic query, it is annotated incorrectly in a way that makes it impossible to find with a syntactic query. If the researcher does this for various instances of the construction, they should get a clear idea of the reliability of the annotation and what sort of errors to look out for. However, it would be impossible to find all annotation errors in this way, since for most research questions it would not be possible to search for every instantiation of a construction.

One disadvantage of this method is that it requires generalization. If you measure the recall for the *hebben gehad* verbal cluster, you might assume that the recall is similar for other two-verb clusters, but this is not necessarily true — perhaps the recall is worse for less frequent words. This concern may be alleviated by sampling a variety of instantiations of the construction. An advantage of the method is that it can be used not only to evaluate the quality of the automatic annotation, but also of the annotation scheme. It has been argued, most notably by Sinclair (2004), that it is better to avoid any sort of annotation if possible, as this already imposes theory upon the data. It may be the case that the annotation scheme of the corpus makes incorrect theoretical assumptions, groups different phenomena together into one category or makes arbitrary distinctions. By performing a query that avoids the annotation altogether and

comparing its results to those of a query that does make use of the annotation, such issues can be detected by a linguistic expert.

I have also applied this method to the Wikipedia data. I could not use only the first 10.000 sentences of the corpus, because there is only one verb cluster instance that occurs more than once in this sample. Instead, I took the first 300.000 sentences of the corpus and searched for the string *hebben gehad*, a common combination of common words, as well as the syntactic version: the verb cluster *hebben gehad*. The syntactic search resulted in four correct examples of *hebben gehad* verb clusters, while the string search provided 14 results, of course including the 4 correct examples from the syntactic search. Nine of the other results were actually verb clusters in main clauses, which are not the target construction, but the distinction cannot be made with just a string search because main clause verb clusters and subordinate clause verb clusters have the same form. However, the remaining string search result was indeed a valid *hebben gehad* verb cluster. On closer examination it had been parsed incorrectly, and therefore it could not have been identified by the syntactic search. It occurs in a sentence with an unusual structure that the parser apparently failed to parse completely, with the main verb *hebben* being left outside of the sentence structure. From this string search, it appears that there were actually five clusters, of which four were identified by the syntactic search. The recall here is 0.8, over this extremely limited sample for this construction.

5. Conclusion

In this paper, I have discusses the issue of using and evaluating linguistic data from automatically annotated treebanks for the purposes of linguistic research. I compared four approaches to evaluation and illustrated them with examples based on a recent linguistic study. These evaluation methods may help to alleviate the concerns that linguists often have about the inaccuracies of such corpora and provide more detail than traditional measures of parsing accuracy when the goal is to study specific constructions.

Since the proposed methods all have different advantages and disadvantages, it would be best to combine them when studying a particular construction. Manual evaluation of the results can be used to determine the precision of a corpus query's results, while searching for particular instances can be used to calculate recall over a portion of the data, determining how many examples might have been missed. Falling back to simpler annotation can be used as a verification of the syntactic annotation of the corpus, even over larger amounts of data, and provide a rough estimate of recall.

While the methods do require some manual annotation effort, they allow a linguistic researcher to get a clearer impression of the quality of the annotation of the particular construction they are investigating in the corpus, while still preserving the advantage of being able to obtain many exemplars with relatively little manual effort. Reporting on such a construction-specific evaluation in a large-scale corpus or treebank study makes the results easier to interpret for those who are not familiar with the errors that an automatic syntactic parser might make. Clearer quantification of the error rate for the linguistic phenomenon that is being studied will also allow researchers to make more definite claims on the basis of data from automatically annotated treebanks. In future work, a larger-scale empirical evaluation of these approaches on a wider variety of constructions and corpora could be conducted to assess them in more detail, and perhaps to create a better reason for linguists to use automatically annotated treebanks in their studies. Furthermore, it may be interesting to investigate whether construction-specific accuracy scores can be incorporated into corpus-based statistical models of language phenomena as part of the margin of error.

Acknowledgements

I would like to thank an anonymous reviewer for their helpful suggestions, and Arjen Versloot and Fred Weerman for the insightful discussions on linguistics and corpora.

6. References

Augustinus, L., Vandeghinste, V., and Van Eynde, F. (2012). Example-based treebank querying. In *Proceedings of the eighth international conference on Language Resources and Evaluation (LREC)*, pp. 3161–3167.

Bastiaanse, R. and Bouma, G. (2007). Frequency and linguistic complexity in agrammatic speech production. *Brain and Language*, 103(1): pp. 78–79.

Bloem, J., Versloot, A., and Weerman, F. (2014). Applying automatically parsed corpora to the study of language variation. In Jan Hajic et al., editors, *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pp. 1974–1984, Dublin, August. Dublin City University and Association for Computational Linguistics.

Bouma, G. and Spenader, J. (2008). The distribution of weak and strong object reflexives in Dutch. *LOT Occasional Series*, 12: pp. 103–114.

Bresnan, J. (2007). Is syntactic knowledge probabilistic? Experiments with the English dative alternation. *Roots: Linguistics in search of its evidential base*, pp. 75–96.

De Sutter, G. (2009). Towards a multivariate model of grammar: The case of word order variation in Dutch clause final verb clusters. In A Dufter, et al., editors, *Describing and Modeling Variation in Grammar*, pp. 225–255. Walter De Gruyter.

Gries, S. T. (2001). A multifactorial analysis of syntactic variation: Particle movement revisited. *Journal of quantitative linguistics*, 8(1): pp. 33–50.

Hinrichs, E. and Beck, K. (2013). Auxiliary fronting in German: A walk in the woods. In *The Twelfth Workshop on Treebanks and Linguistic Theories (TLT12)*, p. 61.

Kilgarriff, A., Rychly, P., Smrz, P., and Tugwell, D. (2004). The Sketch Engine. *Information Technology*, 105: pp. 116.

Lehmann, H. M. and Schneider, G. (2012). Syntactic variation and lexical preference in the dative-shift alternation. *Language and Computers*, 75(1): pp. 65–75.

Lin, Y., Michel, J.-B., Aiden, E. L., Orwant, J., Brockman, W., and Petrov, S. (2012). Syntactic annotations for the

Google Books Ngram corpus. In *Proceedings of the ACL 2012 system demonstrations*, pp. 169–174. Association for Computational Linguistics.

Müller, F. H. (2004). Stylebook for the Tübingen partially parsed corpus of written German (TüPP-D/Z). In *Sonderforschungsbereich 441, Seminar für Sprachwissenschaft, Universität Tübingen*, volume 28.

Napoles, C., Gormley, M., and Van Durme, B. (2012). Annotated Gigaword. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction*, pp. 95–100. Association for Computational Linguistics.

Nivre, J. and Nilsson, J. (2004). Multiword units in syntactic parsing. *Proceedings of Methodologies and Evaluation of Multiword Units in Real-World Applications (MEMURA)*.

Odijk, J. (2015). Linguistic research with PaQu. *Computational Linguistics in The Netherlands journal*, 5: pp. 3–14.

Sinclair, J. (2004). *Trust the text: Language, corpus and discourse*. Routledge.

van der Beek, L., Bouma, G., and van Noord, G. (2002). Een brede computationele grammatica voor het Nederlands. *Nederlandse Taalkunde*, 2002: pp. 353–374.

van Noord, G. and Bouma, G. (2009). Parsed corpora for linguistics. In *Proceedings of the EACL 2009 Workshop on the Interaction between Linguistics and Computational Linguistics: Virtuous, Vicious or Vacuous?*, pp. 33–39. Association for Computational Linguistics.

van Noord, G., Mertens, P., Fairon, C., Dister, A., and Watrin, P. (2006). At Last Parsing Is Now Operational. In *TALN06. Verbum Ex Machina. Actes de la 13e conference sur le traitement automatique des langues naturelles*, pp. 20–42. Leuven University Press.

van Noord, G., Bouma, G., Van Eynde, F., de Kok, D., Linde, J., Schuurman, I., Sang, E. T. K., and Vandeghinste, V. (2013). Large scale syntactic annotation of written Dutch: Lassy. In Peter Spyns et al., editors, *Essential Speech and Language Technology for Dutch*, Theory and Applications of Natural Language Processing, pp. 147–164. Springer Berlin.

van Noord, G. (2009). Huge parsed corpora in LASSY. In F. Van Eynde, et al., editors, *Proceedings of the Seventh International Workshop on Treebanks and Linguistic Theories (TLT 7)*, volume 12, pp. 115–126. LOT.

COPPA V2.0: Corpus Of Parallel Patent Applications Building Large Parallel Corpora with GNU Make

Marcin Junczys-Dowmunt, Bruno Pouliquen, Christophe Mazenc
World Intellectual Property Organization
34, chemin des Colombettes
CH-1211 Geneva 20

{Marcin.Junczys-Dowmunt, Bruno.Pouliquen, Christophe.Mazenc}@wipo.int

Abstract

WIPO seeks to help users and researchers to overcome the language barrier when searching patents published in different languages. Having collected a big multilingual corpus of translated patent applications, WIPO decided to share this corpus in a product called COPPA (Corpus Of Parallel Patent Applications) to stimulate research in Machine Translation and in language tools for patent texts. A first version was released in 2011 but contained only French and English languages. It has been decided to release a major update of this product containing newer data (from 2011 up to 2014) but also other languages (German, English, French, Japanese, Korean, Portuguese, Spanish, Russian and Chinese). This corpus can be used for terminology extraction, cross-language information retrieval or statistical machine translation. With the new version a huge number of files (more than 26 million) has to be processed. We describe the technical process in details.

Keywords: Parallel Corpus of Patents, Build System, GNU Make

1. Introduction

WIPO is a specialized agency of the United Nations dealing with Intellectual Property. WIPO notably administers the Patent Cooperation Treaty (PCT[1]) and while publishing international patent applications, translates the associated titles and abstracts into both English and French. These applications are submitted in one of the PCT publication language (Arabic, German, English, French, Japanese, Korean, Portuguese, Spanish, Russian, or Chinese). Therefore WIPO has an extensive parallel corpus of manually translated patent documents collected over time, especially for the language pair English-French (more than 1.7 million documents), but also from/to other languages (German, Japanese, Korean, Portuguese, Spanish, Russian, or Chinese[2]).

PCT Patent applications are published on the PATENTSCOPE search engine[3], together with other national and international collections. WIPO has investigated techniques for overcoming the language barrier such as cross-language retrieval and machine translation, and developed its own tools based on the open-source toolkit Moses (Koehn et al., 2007), benefiting from academic research results in machine translation.

Cross language Information Retrieval: The fact that WIPO has searchable patent documents in various languages has led to building a tool (called CLIR[4]) to allow users to easily search simultaneously in those various languages.

Statistical Machine Translation: The COPPA corpus has first been fed into an open-source-based statistical machine translation tool (called TAPTA: Translation Assistant for Patent Titles and Abstracts[5]). It can translate texts from English into German, French, Japanese, Korean, Spanish, Russian or Chinese, and vice-versa, (Pouliquen et al., 2011).

In order to further promote research in this field, WIPO decided in 2011 to release the PCT parallel English-French corpus in an easy-to-use TMX format in a product called COPPA (Pouliquen and Mazenc, 2011). However this corpus contained only English and French texts, and it has been decided to extend the corpus with more languages and more recent applications.

2. COPPA: Corpus Of Parallel Patent Applications

The segments included in the corpus are obtained by aligning the sentences of the abstracts and titles of published PCT applications with their translations, the translations having been produced by professional patent translators (More than 200,000 new PCT applications are published every year). It is therefore a gold mine for linguistic research such as terminology extraction, translation memory building and research on Machine Translation.

With the goal of supporting innovation in the Machine Translation field, WIPO offers the updated corpus under the same conditions as before, the product being notably free of charge for academic and private research institutions for research purposes only; in return those institutions commit to share their published results with WIPO.

WIPO hopes that the wide availability of this improved corpus will actively contribute to progress in building more accurate machine translation systems for patent texts with the

[1] Also called PCT application, see WIPO (2010).

[2] Only 25 PCT applications were published in the Arabic language (9/10/2015). We decided for this version not to include them.

[3] http://www.wipo.int/patentscope/search

[4] Publicly available at: https://patentscope.wipo.int/search/en/clir/clir.jsf

[5] Publicly available at https://www3.wipo.int/patentscope/translate

Language pair	Documents	Sentences	Tokens	Characters
en-de	289’287	982’510	36’814’520	225’972’826
en-es	18’303	62’057	2’328’713	14’624’745
en-fr	2’570’292	10’557’032	316’271’950	2’006’750’520
en-ja	312’664	1’036’614	42’127’479	264’578’974
en-ko	41’093	120’534	5’813’474	37’047’347
en-pt	2’001	7’000	261’843	1’696’039
en-ru	6’972	37’261	1’241’791	7’841’040
en-zh	289’287	982’510	36’814’520	225’972’826
Total	3’240’612	12’803’008	404’859’770	2’558’511’491

Table 1: Statistics for the complete corpus. The total does not reflect unique documents as all the documents are available in English and French (a Japanese document - in the en-ja corpus - will also be part of the en-fr subcorpus)

Language		Into English	From English
German	(de)	44.68	30.85
Spanish	(es)	32.97	34.27
French	(fr)	51.06	51.74
Japanese	(jp)	30.54	25.84
Korean	(ko)	25.99	27.95
Russian	(ru)	24.48	32.37
Chinese	(zh)	35.77	32.68

Table 2: BLEU scores for SMT output with the provided test set

ultimate goal of lowering the linguistic barrier for inventors and the general public and of improving the efficiency and the accessibility of the international patent system.

2.1. Statistics

The corpus now contains more than 300 Million words (English-French), for comparison (only for English-French), the previous COPPA version contained 180 Million words, the European corpora (DGT-Acquis/DCEP, (Steinberger et al., 2006)) are about 100 Million words each. See Table 1 for full details.

2.2. Usage in statistical machine translation

We trained our “TAPTA” software on the data provided. The evaluation results are summarized in table 2 (note that the Portuguese COPPA data is too small and has been ignored).

For each language, the new corpus is divided into three distinct sets: a training set (all data until 2014), a development set, and a test set (data taken from early 2015 applications). The training of any statistical models should be done exclusively on the given training set.

Sentences longer than 80 words were discarded. To speed up the word alignment procedure, we split the training corpora into four equally sized parts that are aligned with MGIZA++ (Gao and Vogel, 2008), running 5 iterations of Model 1 and the HMM model on each part.[6] We use a 5-gram language model trained from the target parallel data, with 3-grams or higher order being pruned if they occur only once. Apart from the default configuration with a lexical reordering model, we add a 5-gram operation sequence model (Durrani et al., 2013) (all n-grams pruned if they occur only once) and a 9-gram word-class language model with word-classes produced by word2vec (Mikolov et al., 2013) (3-grams and 4-grams are pruned if they occur only once, 5-grams and 6-grams if they occur only twice, etc.), both trained using KenLM (Heafield et al., 2013). To reduce the phrase-table size, we apply significance pruning (Johnson et al., 2007) and use the compact phrase-table and reordering data structures (Junczys-Dowmunt, 2012). During decoding, we use the cube-pruning algorithm with stack size and cube-pruning pop limits of 1,000.

The development set has been used to tune Moses parameters (using MERT) for the obtained model, while the test set has been used to measure the BLEU scores of the final model. As a result, research teams can use the COPPA corpus in the same conditions, and have a first baseline to benchmark their solution against the BLEU scores obtained by WIPO.

2.3. Technical details

The previous version of COPPA was using the widely used TMX format[7], however we found it more convenient to use TEI[8] for this version and use scripts to export from this format to others. Each document contains, in addition, some meta data that can be extremely useful to use for machine learning: the associated International Patent Classification codes (IPC codes) (which can be used to train “domain-aware” tools as with CLIR and TAPTA),the main applicant’s name, the language of filing (which is a good indication on the direction the translation was done), the application identifier (which also contains the patent office identification) and two dates (application date and publication date).

2.4. Availability

The corpus is available for free for research purposes and for a nominal fee for other purposes, order form and details are available at: `http://www.wipo.int/patentscope/en/data/products.html#coppa`

[6] We confirmed that there seemed to be no quality loss due to splitting and limiting the iterations to simpler alignment models.

[7] `http://www.lisa.org/tmx`

[8] `http://www.tei-c.org`

```
<?xml version="1.0" encoding="utf-8"?>
<TEI.2 id="WO2014071330-fr" lang="fr">
  <teiHeader>
    <fileDesc>
      <titleStmt>
        <title>PROCÉDÉ ET SYSTÈME DE TRAITEMENT DE LANGA
      </titleStmt>
    </fileDesc>
    <notesStmt>
      <note type="ID">WO2014071330</note>
      <note type="AD">20131105</note>
      <note type="ANID">US2013068360</note>
      <note type="DP">20140509</note>
      <note type="IC">G06F 17/28</note>
      <note type="LGF">EN</note>
      <note type="OF">WO</note>
      <note type="PA">FIDO LABS INC.</note>
    </notesStmt>
  </teiHeader>
  <text>
    <body>
      <head id="1" lang="fr">PROCÉDÉ ET SYSTÈME DE TRAIT
      <div type="abstract">
        <p id="2">
          <s id="2:1" lang="fr">La présente invention co
          <s id="2:2" lang="fr">Des modes de réalisation
          <s id="2:3" lang="fr">Pour accroître la précis
          <s id="2:4" lang="fr">Des modes de réalisation
          <s id="2:5" lang="fr">La sortie du LD est faci
          <s id="2:6" lang="fr">La présente invention co
          <s id="2:7" lang="fr">La présente invention co
        </p>
      </div>
    </body>
  </text>
</TEI.2>
```

(a) French example document

```
<?xml version="1.0" encoding="utf-8"?>
<TEI.2 id="WO2014071330-en" lang="en">
  <teiHeader>
    <fileDesc>
      <titleStmt>
        <title>NATURAL LANGUAGE PROCESSING SYSTEM AND ME
      </titleStmt>
    </fileDesc>
    <notesStmt>
      <note type="ID">WO2014071330</note>
      <note type="AD">20131105</note>
      <note type="ANID">US2013068360</note>
      <note type="DP">20140509</note>
      <note type="IC">G06F 17/28</note>
      <note type="LGF">EN</note>
      <note type="OF">WO</note>
      <note type="PA">FIDO LABS INC.</note>
    </notesStmt>
  </teiHeader>
  <text>
    <body>
      <head id="1" lang="en">NATURAL LANGUAGE PROCESSING
      <div type="abstract">
        <p id="2">
          <s id="2:1" lang="en">A natural language proce
          <s id="2:2" lang="en">Embodiments of the NLP s
          <s id="2:3" lang="en">Rules can be added or mo
        </p>
      </div>
    </body>
  </text>
</TEI.2>
```

(b) English example document

```
<linkGrp fromDoc="Xml/fr/WO2014/07/13/WO2014071330.xml"
  toDoc="Xml/en/WO2014/07/13/WO2014071330.xml"
  score="0.158818">
  <link type="1-1" xtargets="1;1" score="1" />
  <link type="1-1" xtargets="2:1;2:1" score="0.239642"/>
  <link type="1-1" xtargets="2:2;2:2" score="0.345575"/>
  <link type="1-1" xtargets="2:3;2:3" score="0.526508"/>
  <link type="0-1" xtargets=";2:4" score="0" />
  <link type="0-1" xtargets=";2:5" score="0" />
  <link type="0-1" xtargets=";2:6" score="0" />
  <link type="0-1" xtargets=";2:7" score="0" />
</linkGrp>
```

(c) Sentence alignment information between two documents

Figure 1: TEI-based XML format of corpus files

3. Creating the Parallel Corpus

During processing, we differentiate between primary and secondary language pairs. Primary language pairs consist of one Non-English language and English. Secondary language pairs are formed from all Non-English languages.

Figure 2 illustrates all processing steps for creating the sentence alignment link file from two parallel documents for a primary language pair, here English-French. The shown dependency graph is modeled very closely after our pipeline based on GNU Make.

After converting binary formats (MS Word, WordPerfect) to the presented TEI-XML format, sentence splitting[9] is applied to the XML-file, retaining the original paragraph structure as shown in Figure 1a.

To ensure a high sentence alignment quality, we rely on a two-step approach similar to (Sennrich and Volk, 2011). French documents are translated into English first. We randomly select a subset of 10,000 document pairs and align them using Hunalign (Varga et al., 2005), selecting only 1-1 alignments that are themselves surrounded by 1-1 alignments. This small lower-quality parallel corpus is used to train an SMT system with Moses (Koehn et al., 2007). Following (Sennrich and Volk, 2011) we use significance pruning (Johnson et al., 2007) to filter out noise resulting from alignment errors.

Next, our monolingual sentence aligner BLEU-Champ[10] is applied. BLEU-Champ relies on smoothed sentence level BLEU-2 as a similarity metric between sentences and uses the Champollion algorithm (Ma, 2006) with that metric. To avoid computational bottlenecks for long documents, first

[9] Using Eserix, an SRX-based sentence splitter `https://github/emjotde/eserix`. The algorithm and rules have been extracted from Psi-Toolkit (Graliński et al., 2012).

[10] `https://github/emjotde/bleu-champ`

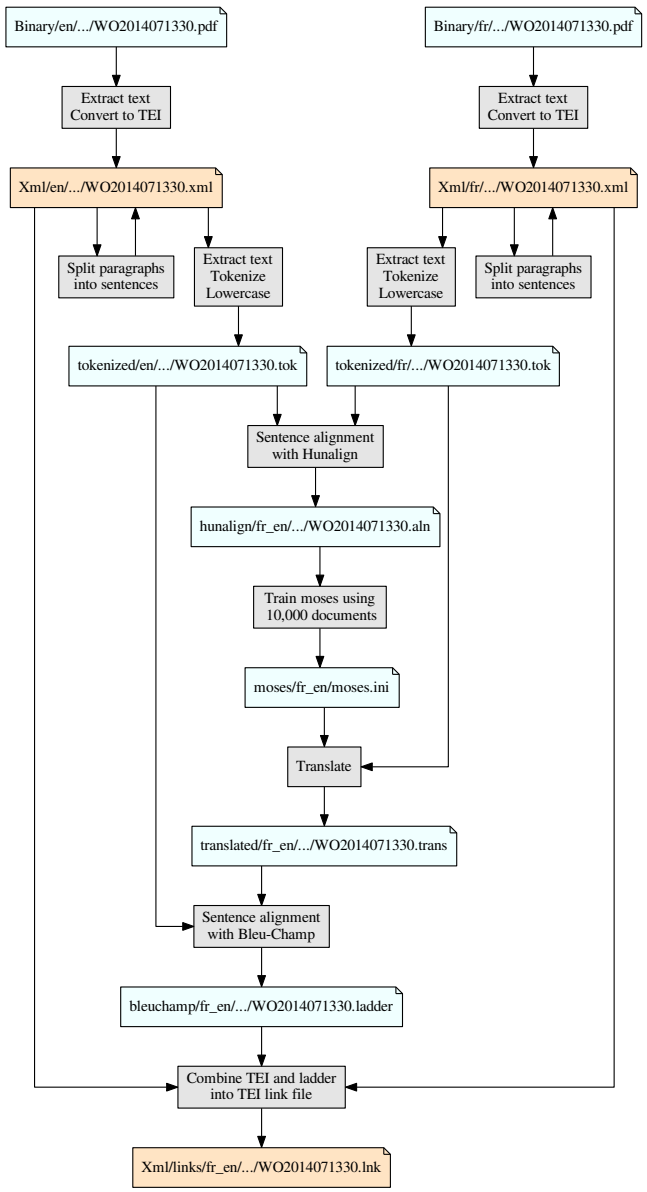

Figure 2: GNU Make dependencies for sentence alignment procedure

a path consisting only of 0-1, 1-0, 1-1 alignments is calculated. In a second step, the search is restricted to a 10-sentence-wide corridor around the best path allowing for all alignment combinations up to 4-4 alignments. This procedure avoids search errors and is fast enough to use the Champollion algorithm with documents consisting of thousands of sentences. Given the English tokenized text and the translated French text, BLEU-Champ produces a ladder file (Hunalign's numeric sentence alignment format) which in the end is combined with the two TEI documents to form the final TEI sentence alignment file (see 1c).

The beige-colored TEI files in Figure 2 are distributed as part of the corpus. Since the link files contain pointers to the original XML documents any set of link files can be used to produce plain-text parallel corpora.

In case of secondary language pairs, the steps are the same with the exception that both documents are translated into English and sentence alignment is performed on the English translation results of both files.

The entire process creates 9,373,728 XML files (document files and link files) meant for distribution and 17,065,732 temporary intermediate targets (plain text tokenized, translated files). Thanks to the use of GNU Make, we can parallelize the processing across 64 physical cores taking advantage of the full available computational power of the used machine. Occasional crashes or interruptions are no problem as the system can easily resume work with minimal overhead.

4. Conclusions

One of the mandates of WIPO is to facilitate access to technical knowledge and information. To achieve this goal, WIPO encourages innovation by providing its corpus of translated patent application (COPPA) free of charge for research purposes.

Our baselines and test sets can serve as reference data for future publications and we would like researchers to explore machine translation techniques beyond the phrase-based approach that was used to produce them. The meta-information and preserved document structure provided can help to advance recent work in document-level translation.

By choosing GNU Make as a build system for our corpus, we created a self-updating processing chain that allows us to easily add new documents with optimal processing steps. By this we can maintain current versions of the corpus and prepare them with minimal effort for possible future updates. The automatic parallelization of GNU Make made it possible to process millions of files in a relatively short time.

5. References

Durrani, N., Fraser, A., Schmid, H., Hoang, H., and Koehn, P. (2013). Can Markov models over minimal translation units help phrase-based SMT? In *ACL*, pages 399–405. The Association for Computer Linguistics.

Gao, Q. and Vogel, S. (2008). Parallel implementations of word alignment tool. In *Software Engineering, Testing, and Quality Assurance for Natural Language Processing*, pages 49–57. ACL.

Graliński, F., Jassem, K., and Junczys-Dowmunt, M. (2012). PSI-Toolkit: Natural Language Processing Pipeline. *Computational Linguistics - Applications*, pages 27–39.

Heafield, K., Pouzyrevsky, I., Clark, J. H., and Koehn, P. (2013). Scalable modified Kneser-Ney language model estimation. In *Proceedings of the 51st Annual Meeting of the ACL*, pages 690–696.

Johnson, J. H., Martin, J., Forst, G., and Kuhn, R. (2007). Improving translation quality by discarding most of the phrasetable. In *In Proceedings of EMNLP-CoNLL?07*, pages 967–975.

Junczys-Dowmunt, M. (2012). Phrasal Rank-Encoding: Exploiting phrase redundancy and translational relations for phrase table compression. *Prague Bull. Math. Linguistics*, 98:63–74.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran,

C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E. (2007). Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, pages 177–180, Stroudsburg, USA. Association for Computational Linguistics.

Ma, X. (2006). Champollion: A robust parallel text sentence aligner. In *In Proceedings of LREC-2006*.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

Pouliquen, B. and Mazenc, C. (2011). COPPA, CLIR and TAPTA: three tools to assist in overcoming the patent barrier at WIPO. In *Proc of MT-Summit XIII*, pages 24–30, Xiamen, China.

Pouliquen, B., Mazenc, C., and Iorio, A. (2011). Tapta: A user-driven translation system for patent documents based on domain-aware statistical machine translation. In Mikel L. Forcada, et al., editors, *Proceedings of th 15th International Conference of the European Association for Machine Translation (EAMT)*, pages 5–12.

Sennrich, R. and Volk, M. (2011). Iterative, mt-based sentence alignment of parallel texts. *18th Nordic Conference of Computational Linguistics, NODALIDA*.

Steinberger, R., Pouliquen, B., Widiger, A., Ignat, C., Erjavec, T., and Tufiş, D. (2006). The JRC-Acquis: A multilingual aligned parallel corpus with 20+ languages. In *Proceedings of the 5th International Conference on Language Resources and Evaluation*, pages 2142–2147.

Varga, D., Németh, L., Halácsy, P., Kornai, A., Trón, V., and Nagy, V. (2005). Parallel corpora for medium density languages. In *Recent Advances in Natural Language Processing (RANLP 2005)*, pages 590–596.

Efficient Exploration of Translation Variants in Large Multiparallel Corpora Using a Relational Database

Johannes Graën, Simon Clematide, Martin Volk
Institute of Computational Linguistics
University of Zurich
Zurich, Switzerland
{graen|siclemat|volk}@cl.uzh.ch

Abstract

We present an approach for searching and exploring translation variants of multi-word units in large multiparallel corpora based on a relational database management system. Our web-based application Multilingwis, which allows for multilingual lookups of phrases and words in English, French, German, Italian and Spanish, is of interest to anybody who wants to quickly compare expressions across several languages, such as language learners without linguistic knowledge.

In this paper, we focus on the technical aspects of how to represent and efficiently retrieve all occurrences that match the user's query in one of five languages simultaneously with their translations into the other four languages. In order to identify such translations in our corpus of 220 million tokens in total, we use statistical sentence and word alignment.

By using materialized views, composite indexes, and pre-planned search functions, our relational database management system handles large result sets with only moderate requirements to the underlying hardware. As our systematic evaluation on 200 search terms per language shows, we can achieve retrieval times below 1 second in 75 % of the cases for multi-word expressions.

Keywords: corpora, multiparallel, retrieval, database, evaluation

1. Introduction

In recent years, large parallel corpora have become popular not only for natural language processing but also for linguistic research and for language learners. Arguably, the most popular site is Linguee[1] which offers bilingual lexicon searches in combination with usage examples over word-aligned parallel corpora. These online systems have a number of shortcomings (Volk, Graën, and Callegaro, 2014). Most notably, they are restricted to bilingual searches. If a user is interested in a multilingual comparison, she must submit multiple queries.

On that account, we are developing a new corpus exploration tool to investigate translation variants in large multiparallel corpora. Our system *Multilingwis*[2] *(Multilingual Word Information System)* contains the texts of five languages from *Europarl*[3] with cross-language alignments down to the word level. Multilingwis allows the user to search for single words or multi-word expressions and returns the corresponding translation variants in the four other languages. Translation variants are all words and phrases that result from our statistical word alignment.

Corpus search systems for expert users require linguistic knowledge and information about the annotation layers, e.g. morphological symbols, part-of-speech tags, grammatical categories or how to infer the lemma given a word in a particular language. On the contrary, Multilingwis follows the principle of strict simplicity. The user types any word sequence as a query which is then interpreted by the system. First, the system determines the most likely language of the input words based on frequencies learned from the corpus. Then it strips the input sequence of all function words and triggers the query with the lemmas of all content words (adjectives, adverbs, nouns and verbs). Multilingwis retrieves all sentences with the search words in the given order where there are three or less function words in between any two search words. The challenge then lies in finding and highlighting the corresponding hits in the four target languages efficiently.

This paper first describes the preparation and linguistic annotation of our multiparallel corpus which is based on Europarl. We then describe in detail our technical solution for efficient retrieval based on advanced database techniques. Our evaluation shows that multiword retrieval for high-frequency input terms can be done efficiently even on large data sets.

2. Corpus Preparation

We extracted parallel text units[4] in English, French, German, Italian and Spanish from the *Corrected & Structured Europarl Corpus (CoStEP)*[5] (Graën, Batinic, and Volk, 2014) to each of which we subsequently applied the *TreeTagger* (Schmid, 1994) for tokenization, part-of-speech tagging and lemmatization. Tagging was done with the language models available from the TreeTagger's web page[6]. We adapted the TreeTagger's tokenizer (abbreviation lexicons, punctuation) and extended its tagging lexicon (especially the German one) with lemmas and part-of-speech tags for frequent words unknown to the language models.

[1] `http://www.linguee.com/`

[2] `http://pub.cl.uzh.ch/purl/multilingwis`

[3] `http://www.statmt.org/europarl/`

[4] Here, speaker turns from the sittings of the European Parliament.

[5] Altogether 146,652 speaker turns are available in all these five languages in CoStEP version 0.9.2, which bases on *Europarl* release v7 (Koehn, 2005). CoStEP is available at `http://pub.cl.uzh.ch/purl/costep/`.

[6] `http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/#Linux`

Language-specific rules based on word forms, lemmas and part-of-speech tags allowed us to identify sentence segment boundaries, which separate parts of sentences by colon or semicolon[7]. After identifying sentence segments (about 1.7 million per language), we performed pairwise sentence alignment with *hunalign* (Varga et al., 2005) and based on that word alignment with *GIZA++* (Och and Ney, 2003; Gao and Vogel, 2008). Word alignment was performed on the lemmas[8] of content words for both directions on each language pair.

Having the corpus data processed as detailed above, we stored the data in a relational database as described in Graën and Clematide (2015). Our relational database management system (RDBMS) of choice is PostgreSQL[9] as it provides all the functionality that we rely on for our application.[10]

3. Efficient Retrieval from the Corpus Database

Since our retrieval method relies on lemmas both for source (query) and target (translation) languages, we built a *materialized view*[11] on lemmas including all relevant foreign keys so that this view comprises all relevant data and can be queried later on instead of the underlying tables. In case no lemma was given for a particular token[12], we include the word form instead as aforementioned. The view comprises one row (i.e. lemma tuple) for each original token which sums up to 220 millions over all languages and corresponds roughly to 44 million tokens per language.

We then built a *composite index* (see Winand, 2012, pp. 12–17) upon that view starting with the lemma itself and including all other columns in the order accessed by the query (lemma index) with the objective of not needing to fetch any actual data but the index when performing a corpus search. The index requires 7.3 GB of disk space which only adds 2.2 GB compared to an ordinary index over the lemma attribute of all 220 million rows.

In addition, we created a composite index on a symmetrized view of the word alignments (alignment index) that we had calculated. As symmetrization method we chose the union (Tiedemann, 2011, p. 76), thus favoring recall for our application. This index comprises 418 million single word alignments and requires 9 GB of disk space.

The search query first scans the lemma index in order to retrieve all matching token tuples within the same sentence segments for the search terms given. It then looks up the aligned tokens in all other languages by consulting the alignment index. Since we are not interested in the exact correspondence of lemmas from source to target languages but rather in the corresponding list of lemmas ordered by their appearance in the text, we can use the token tuple from the source language as a set when consulting the alignment index and hence the index gets scanned only once.

Subsequently, the query makes use of the lemma index again to retrieve lemmas for the tokens aligned which are concatenated to identify the particular *translation variant*.

For every reasonable count of search terms (up to nine words), we created a particular search function in the database in order for the database's query planner to already have a query plan (see Winand, 2012, pp. 172–179) prepared and, thus not needing to deal with it at runtime. Using several search functions, each one addressing a specific count of search terms, considerably outperforms a single function based on *Common Table Expressions (CTE)*[13] or recursion with a list of search terms as input.

Within these functions, we also count the frequencies of translation variants and rank the matching sentence segments of source and target languages by calculating a score that favors consistently short segments in all languages. These ones will be shown first in the example panel of the web application, depending on the user's selection of translation variants.

4. User-friendly Interface

We decided to build Multilingwis with a configuration-free web-based user interface. Upon entering one or more search terms, the system immediately gives feedback on the identified language and the accepted vs. ignored input words (i.e. content vs. function words). The query results appear quickly in the four other languages. They are sorted according to frequency and offer a number of options for the corpus exploration. See Clematide, Graën, and Volk (2016) for a description of the user interface.

In principle, every corpus sentence in the result set can be inspected. For many queries this is impractical because of the large number of hits. Therefore, Multilingwis allows the user to restrict the inspection to combinations of translation variants across languages. Given a German query, for example, the user may restrict the exploration to certain English and Spanish translation variants in combination. Particular variants for each language are hidden if they appear considerably less frequent than the most frequent variant, though the complete list can be checked by unfolding it.

Multilingwis helps investigating lexical variants in a single language by switching queries between languages. For instance, a German query results in a number of Spanish translation variants. By selecting one of those variants as a new query, one will get alternatives to the original German query. In this way, the languages may serve as mirrors for each other.

[7] More than 6 % of the segments in our corpus end with colon or semicolon.

[8] We used the word form instead if no lemma was provided; ambiguous lemmas are not disambiguated.

[9] `http://www.postgresql.org/`

[10] For a detailed feature comparison of major SQL databases see Winand (2012).

[11] Unlike regular views, materialized views are precalculated and thus provide faster access to the data queried in trade for disk space.

[12] These are mostly nouns that are unknown to the TreeTagger model.

[13] So called *Recursive Common Table Expressions* (they are not recursive themselves but their result sets can be understood as recursively defined) are a common way to iterate through list parameters. For our requirement, i.e. finding sentence segments given a list of search terms, a CTE would generate a first set of segments matching the first term and then incrementally build subsets of the respectively anterior set for every subsequent term in the list.

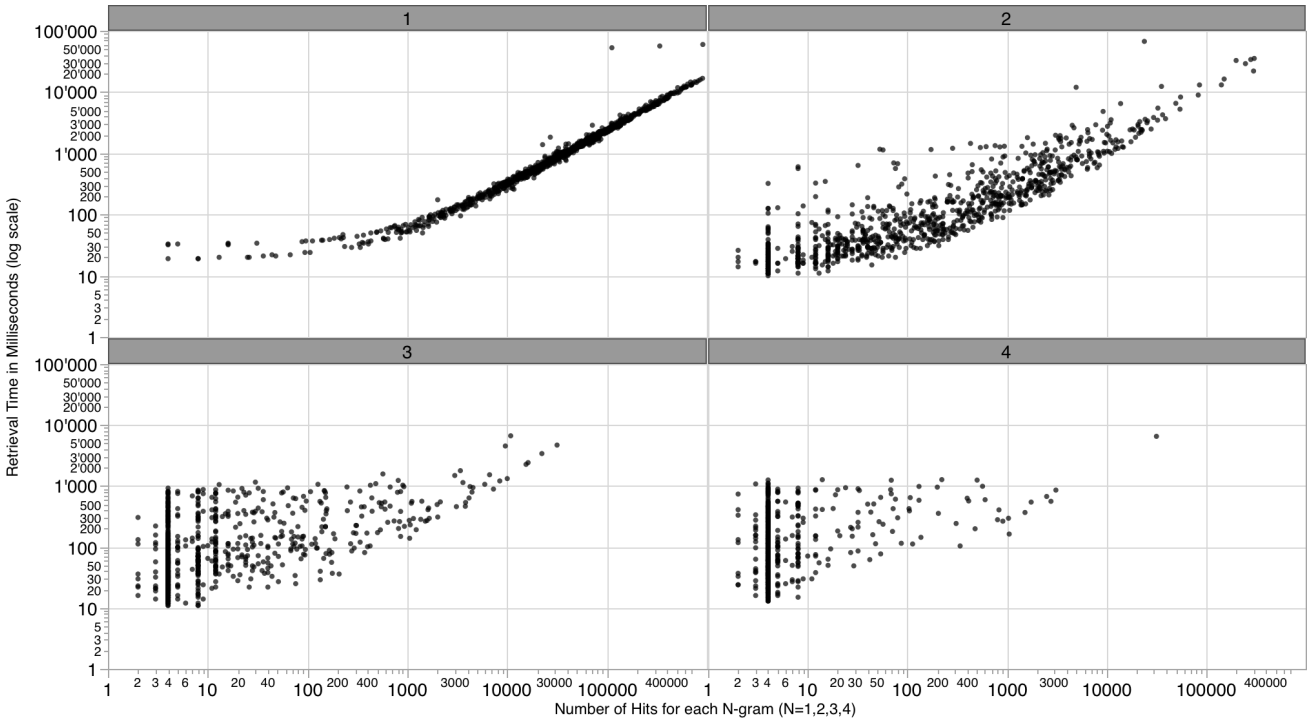

Figure 3: Correlation of the number of translation variants and retrieval time grouped per N-gram (N=1,2,3,4)

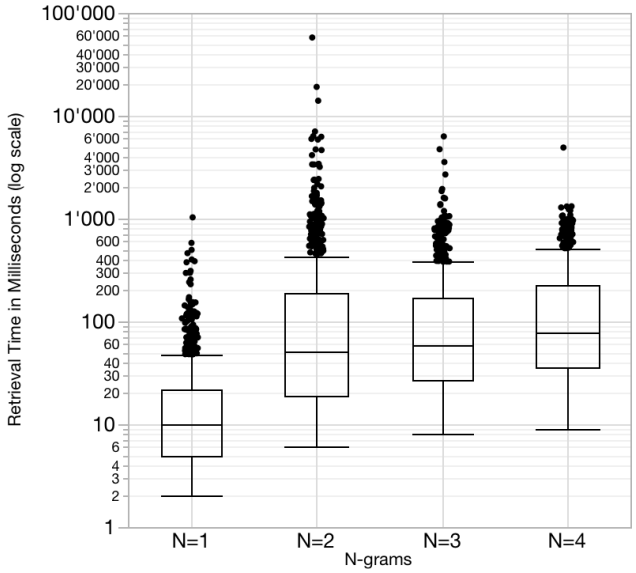

Figure 1: Boxplots of the retrieval time (ms) of all hits in the language of the query

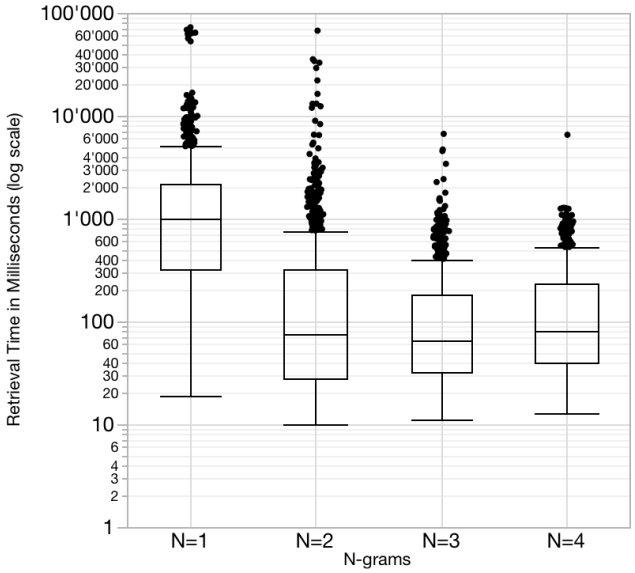

Figure 2: Boxplots of the retrieval time (ms) of all translation variants

5. Evaluation

In order to systematically evaluate the retrieval times of the database queries, we randomly sampled 200 different content lemmas from each language. These lemmas had to be followed by three additional content lemmas in the same sentence allowing for at most three intervening non-content words (proximity windows) between each content word. This experimental setup allows us to evaluate the retrieval times for n-grams of content words which share a common prefix, and, therefore, to assess whether the retrieval time for multi-word units decreases according to their frequency although each element of the multi-word unit might have a high frequency on its own.

All retrieval times were measured by a local PostgreSQL client performing the search on a dedicated Linux database host with PostgreSQL 9.5.0 (Intel Xeon E5-2650 2.6 GHz processors, SSDs for tablespace, 265 GB RAM). The numbers discussed report the time needed for retrieving the number of result rows (`SELECT count(*) FROM …`). For frequent words, the actual retrieval of the resulting rows

(`SELECT * FROM …`) can easily dominate the time needed for calculating that number.

Our first evaluation measures the time needed to find all hits in the language of the query. The boxplots in Fig. 1 show that the 75th percentile value is around 0.5 seconds or less for all languages. However, there are some outliers for combinations of frequent words where the retrieval time may take several seconds.

A further evaluation reports the time needed to retrieve all translation variants of all hits for a query, including the time to retrieve the hits in the language of the query. The boxplots in Fig. 2 show that the retrieval time for 4-gram multi-word units is dominated by the retrieval of the hits in the language of the query. For 4-grams, there are only a few hits in one language, and their translation variants can be found quickly. For 1-grams (single words), a substantial amount of computing time is needed in order to find all translation variants (up to 72 seconds for the highly frequent English verb "be"). However, the 75th percentile retrieval time for multi-word units is still below 1 second. As can be seen in Fig. 3, the correlation between the retrieval time and the number of translations decreases when the N of N-grams increases.

6. Conclusions

We implemented a corpus query system dedicated to the exploration of multi-word units in large multiparallel corpora based on a relational database management system (PostgreSQL).

In this paper, we discussed the technical implementation we chose in order to allow for an efficient retrieval of all translation variants for a given multi-word unit. Database indexes that are geared to the actual queries play a central role for fast retrieval.

Our evaluation shows that most multi-word queries (75 %) can be responded to within less than 1 second. Furthermore, the query response time decreases as the amount of words constituting the multi-word units increases.

7. Acknowledgment

This research was supported by the Swiss National Science Foundation under grant 105215_146781/1 through the project "SPARCLING – Large-scale Annotation and Alignment of Parallel Corpora for the Investigation of Linguistic Variation".

8. References

Clematide, S., J. Graën, and M. Volk (2016). "Multilingwis – A Multilingual Search Tool for Multi-Word Units in Multiparallel Corpora". In: *Computerised and Corpus-based Approaches to Phraseology: Monolingual and Multilingual Perspectives – Fraseología computacional y basada en corpus: perspectivas monolingües y multilingües*. Ed. by G. C. Pastor. Geneva: Tradulex, pp. 447–455.

Gao, Q. and S. Vogel (2008). "Parallel implementations of word alignment tool". In: *Software Engineering, Testing, and Quality Assurance for Natural Language Processing*. Association for Computational Linguistics, pp. 49–57.

Graën, J., D. Batinic, and M. Volk (2014). "Cleaning the Europarl Corpus for Linguistic Applications". In: *Proceedings of the 12th KONVENS*. (Hildesheim), pp. 222–227.

Graën, J. and S. Clematide (2015). "Challenges in the Alignment, Management and Exploitation of Large and Richly Annotated Multi-Parallel Corpora". In: *3rd Workshop on the Challenges in the Management of Large Corpora*. (Lancaster). Ed. by P. Bański, H. Biber, E. Breiteneder, M. Kupietz, H. Lüngen, and A. Witt. Institut für Deutsche Sprache, pp. 15–20.

Koehn, P. (2005). "Europarl: A parallel corpus for statistical machine translation". In: *Machine Translation Summit*. (Phuket). Vol. 5. Asia-Pacific Association for Machine Translation (AAMT), pp. 79–86.

Och, F. J. and H. Ney (2003). "A Systematic Comparison of Various Statistical Alignment Models". In: *Computational linguistics* 29.1, pp. 19–51.

Schmid, H. (1994). "Probabilistic part-of-speech tagging using decision trees". In: *Proceedings of International Conference on New Methods in Natural Language Processing (NeMLaP)*. (Manchester). Vol. 12, pp. 44–49.

Tiedemann, J. (2011). "Bitext Alignment". In: *Synthesis Lectures on Human Language Technologies* 4.2, pp. 1–165.

Varga, D., P. Halácsy, A. Kornai, V. Nagy, L. Németh, and V. Trón (2005). "Parallel corpora for medium density languages". In: *Proceedings of the Recent Advances in Natural Language Processing (RANLP)*. (Borovets), pp. 590–596.

Volk, M., J. Graën, and E. Callegaro (2014). "Innovations in Parallel Corpus Search Tools". In: *Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC)*. (Reykjavik). European Language Resources Association (ELRA), pp. 3172–3178.

Winand, M. (2012). *SQL Performance Explained: Everything Developers Need to Know about SQL Performance*. Markus Winand.

Collection and Indexation of Tweets with a Geographical Focus

Adrien Barbaresi
Institute for Corpus Linguistics and Text Technology
Austrian Academy of Sciences
Sonnenfelsgasse 19 – 1010 Vienna
adrien.barbaresi@oeaw.ac.at

Abstract

This paper introduces a Twitter corpus currently focused geographically in order to (1) test selection and collection processes for a given region and (2) find a suitable database to query, filter, and visualize the tweets. Due to access restrictions, it is not possible to retrieve all available tweets, which is why corpus construction implies a series of decisions described below. The corpus focuses on Austrian users, as data collection grounds on a two-tier detection process addressing corpus construction and user location issues. The emphasis lies on short messages whose sender mentions a place in Austria as his/her hometown or tweets from places located in Austria. The resulting user base is then queried and enlarged using focused crawling and random sampling, so that the corpus is refined and completed in the way of a monitor corpus. Its current volume is 21.7 million tweets from approximately 125,000 users. The tweets are indexed using Elasticsearch and queried via the Kibana frontend, which allows for queries on metadata as well as for the visualization of geolocalized tweets (currently about 3.3% of the collection).

Keywords: Computer-Mediated Communication, Web Corpus Construction, Database Solutions, Visualization

1. Introduction

The availability and ease of use has made the online social networking service Twitter one of the most popular data sources for studying social communication (Leetaru et al., 2013). Generally, the interest in Twitter is considered to reside in the immediacy of the information presented, the volume and variability of the data contained, and the presence of geolocated messages (Krishnamurthy et al., 2008). Other social networks do not deliver the same amount of text, especially for German (Barbaresi, 2015b), and more importantly, cannot be deemed as stable in time in terms of popularity and API access (Barbaresi, 2013).

Short messages published on social networks constitute a "frontier" area due to their dissimilarity with existing corpora (Lui and Baldwin, 2014), most notably with reference corpora. Since August 2009, Twitter has allowed tweets to include geographic metadata (Stone, 2009), which are considered to be a valuable source for performing linguistic studies with a high level of granularity, e.g. on language variation (Ruiz Tinoco, 2013). Thus, from the point of view of corpus and computational linguistics, Twitter data are both highly relevant and difficult to process.

Due to access restrictions, mostly mechanical constraints on the API, it is not possible to retrieve all tweets one would need. For example, when using the so-called "gardenhose" streaming API, it is necessary to enter search terms or a geographic window, and a fraction of corresponding data is returned, which may greatly affect results (Morstatter et al., 2013), especially for highly frequent keywords as used by the TweetCat approach (Ljubešic et al., 2014) or for the *German Twitter Snapshot* (Scheffler, 2014). In that sense, focusing on a given geographical region can be a way to provide enough relevant linguistic evidence. However, there are structural characteristics which complicate the collection of tweets from German-speaking countries, and especially Austria, which makes it a interesting test case.

First, even without considering the market penetration of Twitter, the population of the country is comparatively small, so that Austrian users cannot be expected to be easily found at random, all the more since users preferentially connect to other users from their own country (Kulshrestha et al., 2012). Second, geolocated tweets are a small minority, with estimates as low as 2% of all tweets (Leetaru et al., 2013). Third, because of privacy concerns Austrian users can be expected to be very cautious about geolocation services: German twitterers for example are very reluctant to include geographic coordinates in their tweets (Scheffler et al., 2014). Finally, the success at being able to place users within a geographic region varies with the peculiarities of the region (Graham et al., 2014).

2. Design decisions

Following the characteristics stated above, and because corpus construction in the linguistic tradition implies a number of decisions which have to be made explicit (Barbaresi, 2015a), salient methodological issues will be dealt with in detail in this section.

First, while most studies ground on a collection process which is limited in time, the corpus described in this article is a monitor corpus in the sense that it grows constantly with time. Since metadata include the time of posting, it is possible to split the corpus in units of time. More generally, the purpose is to be opportunistic enough during corpus creation in order to enable researchers to tailor subcorpora which match particular interests.

Second, geolocated tweets (*place* element in the JSON response) may be casually sent from Austria, but not really by Austrian users: they can merely be an indication that the user has spent some time in Austria. Furthermore, it is technically possible to spoof one's location either by editing by hand the location field of a given tweet, or by tampering with the GPS device used for geolocation. On the other hand, the field which is sent with each tweet along with the user profile (*user/location* field), if given, refers to the subjective point of view of the users as regards their lo-

cation. It may not seem as objective as mere coordinates, and even when both the profile and the device location are valid, they do not always correspond (Graham et al., 2014) but it is a strong assertion regarding the place users feel at home or related to at least. Here lies the difference between a mere "posted from Austria" predicate and the corpus construction process which leads to tweets hopefully "made in Austria".

Third, since language cannot reliably be used as a proxy for location (Graham et al., 2014), no language selection is undertaken. For the same reason, retweets are included, even if the original messages may have been posted from other locations and in another context, because they are still considered to be meaningful. They can be removed for further studies by using the metadata as well as the "RT" mentions in the messages (Ruiz Tinoco, 2013). Furthermore, the use of typical Austrian-German words do not seem to lead to a substantial amount of users, due to the mobility of users and due to the difficulty to define a "national variety" (Ebner, 2008), which separates this case from languages like Croatian or Slovene (Ljubešic et al., 2014).

Fourth, geocoding algorithms can be used to help recreate absent geolocation metadata, using textual mentions of place (Leetaru et al., 2013) or linguistic cues (Scheffler et al., 2014) based on the identification of "local words" (Cheng et al., 2010). On the one hand, there are potential ambiguities in place names that have to be resolved to establish a reliable list of Austrian places, which implies a significant amount of work with an unknown outcome. On the other hand, the tweets are not exclusively in German and I do not agree with the segmentation of Austria in one bloc as used by (Scheffler et al., 2014). That is why no attempt is undertaken to recreate location metadata.

Finally, so-called "heavy tweeters" (Krishnamurthy et al., 2008) as well as peculiarities of the API (Morstatter et al., 2013) raise the question of sampling processes. Although human users usually entertain a stable amount of stable relationships (Gonçalves et al., 2011), it is conceivable that heavy users as well as machine-generated tweets account for distortions in the corpus. Additionally, the random sampling methodology used by Twitter to generate the streams of tweets is rarely put into question (Zafar et al., 2015). This means that steps have to be taken in order to minimize the impact of differences in user activity as well as potentially unknown sampling biases.

3. Implementation

To sum up the methodological concerns, what is needed is a method allowing to find and collect tweets from Austrian users with a reasonable precision. My method uses different modules as presented in figure 1. The first component can be considered to be a "lurker" module in the sense that it merely listens to the Twitter's streaming API[1] to collect geolocated tweets whose coordinates are in or close to Austria. Tweets featuring geolocation in Austria or with a user profile location field linked to Austria are singled out. The corresponding user names are then passed to a second module which fetches user streams in order to analyze them. Additionally, the social networks (friends and followers) are crawled (Kumar et al., 2014) in order to find other potentially interesting users, which makes the operation comparable to an API-side focused or scoped crawling (Olston and Najork, 2010). The communication with the API relies on the Python wrapper *twython*.[2]

The constant filtering is meant to optimize the collection. In fact, there are mechanical constraints on both ends: access to the API on user level is limited to 180 requests per slot of 15 minutes, and on the other side unneeded content may clutter up storage devices. Additionally, I found that potentially interesting users are geographically and linguistically very mobile; they may use several languages and be tied to several home places. Finally, even among users who use geolocation services, the proportion of tweets with actual location data may greatly vary, so that users are unequally productive in this respect.

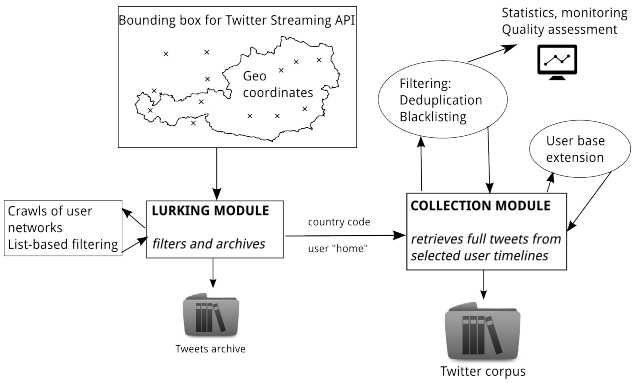

Figure 1: Schema of the implementation

Studies have shown that it is desirable to gather a set of users which is both large and diverse (Zafar et al., 2015), so that the collection process is opportunistic despite a rather conservative setting concerning location: at least 50% of geolocated tweets per user have to be in Austria. Positives in the user location field are found on token level using a fixed list of case-insensitive cues: nationwide mentions (e.g. Austria), all regions (*Bundesländer*), well-known landscapes (e.g. *Waldviertel*), and top-20 cities. For the sake of completeness, the main quarters of the major cities (e.g. *Josefstadt* in Vienna) as well as major geographical features (e.g. valleys and rivers) have been added, however they seem to be rarely used. Quantitatively speaking, the number of users found that way (around 125,000) is concordant with results from market studies, with an estimated number of 140,800 Austrian users in September 2015.[3]

The corpus is constantly growing, and so is the user base. Filtering steps include the deduplication of tweets and the blacklisting of unwanted users, which both yield statistical information for quality assessment. At the same time, remaining tweets are scanned for other user names in replies or retweets, whose timelines are retrieved and stored if they match the location criteria. In order to avoid bias by heavy twitterers, the timelines are fetched at random intervals among the range of valid users.

[1] https://dev.twitter.com/streaming/overview

[2] https://github.com/ryanmcgrath/twython

[3] http://de.statista.com/statistik/daten/studie/296135/umfrage/twitter-nutzer-in-oesterreich/

4. Indexation and results

To keep up with the growing amount of tweets, a specific search engine has been chosen. The interest of NoSQL databases to deal with the feature-rich content return by the Twitter API is known (Kumar et al., 2014). Two main components of the open-source *ELK* stack (Elasticsearch, Logstash, Kibana) are used, namely Elasticsearch[4] to index the tweets and Kibana[5] to provide a user-friendly interface to queries, results, and visualizations. The main drawbacks result at the time being from the lack of linguistic processing: a rather unprecise lemmatization of queries and results by the search engine as well as a lack of linguistic annotation. These tasks will require a substantial amount of testing due to the multiple languages and the difficulty of twitter messages.

Although it is not primarily a search engine for linguists, Elasticsearch takes advantage of the native JSON format of the tweets as well as of a number of relevant field types after a subsequent mapping, which allows for refined queries on text and metadata, for instance "the *-erl* diminutive form in tweets from users with more than 10 followers and with the city of Klagenfurt mentioned in the home location field". In the current implementation, using Kibana's syntax, this query translates to *text:*erl AND user.followers_count:[10 TO *] AND user.location:Klagenfurt*. In order to give a user-friendly access to the results, dashboards can be configurated out of a series of indicators (see figure 2).

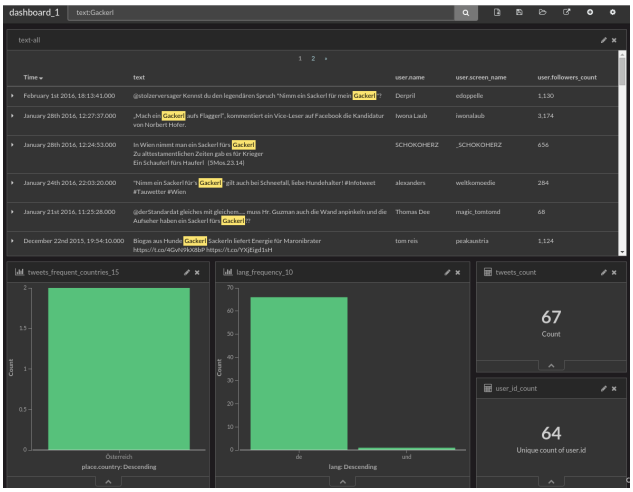

Figure 2: Example of dashboard view

The most frequent languages according to the metadata delivered by Twitter are English (42.2% of all tweets) and German (40.5%), with a number of less frequently represented languages such as Turkish (2.8%), Spanish (1.3%), and Japanese (0.9%). The amount of tweets whose language could not be determined by Twitter is relatively low (6.5%), which indirectly yields insights on the quality of the corpus. This information is confirmed by the mean length of the tweets (100.4 characters and 12.7 tokens).

[4]https://www.elastic.co/products/elasticsearch

[5]https://www.elastic.co/products/kibana

The proportion of geolocated tweets (3.3%) is better than in the comparable *German Snapshot* (Scheffler, 2014), where it amounts to 1.1%. Their distribution by country is largely in favor of Austria (75.0% of geolocated tweets), with a number of other less prominent countries such as the USA (6.2%), Germany (4.1%), and Turkey (1.6%). These figures show that it is necessary to target Austria in comparison to a general approach targeting German. Visualizations of geographical data can be constructed "out of the box" as soon as coordinates have been mapped as geographical data in the database, which allows for the projection of geolocated tweets on a map.

A heat map centered on Austria is shown in figure 3. The distribution of tweets is mostly in line with population distribution, with the exception of Klagenfurt. It highlights the prominence of Vienna and its airport as well as the importance of commuters and travellers, with train tracks partially visible. Holiday resorts such as ski stations are also depicted on the map, which altogether prompts for geographical and sociological analyses of mobility.

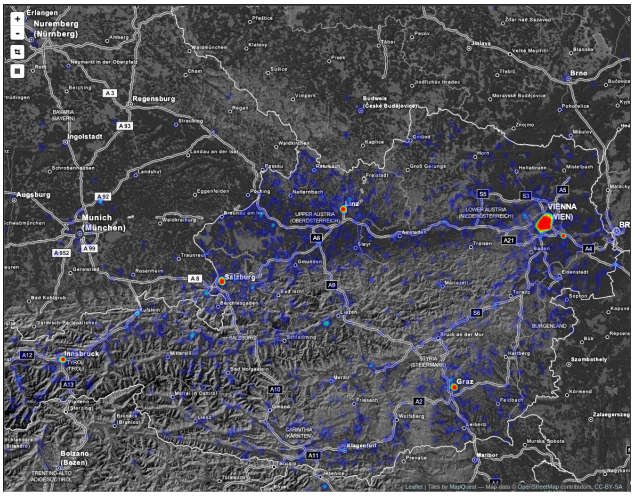

Figure 3: Heat map of all geolocated tweets

5. Conclusion

I introduced a monitor corpus of tweets from Austrian users. The data collection grounds on a two-tier detection process addressing corpus construction and user location issues. The emphasis lies on short messages whose sender (1) mentions a place in Austria as his/her hometown or (2) often tweets from places located in Austria. The resulting user base is then queried and enlarged using random sampling. The current volume of the corpus is 21.7 million tweets from approximately 125,000 users, which is roughly comparable to the *German Snapshot* (Scheffler, 2014) in terms of volume with a number of users one order of magnitude smaller. The tweets are mainly written in English and German. The proportion of geolocated tweets is 3.3%, 75.0% of which come from Austria.

Future work includes work on fine-grained differences in geolocations which could improve the quantitative throughput as well as the qualitative value of the corpus. In the

same perspective, ambiguities of gazetteers have to be reduced to a minimum in order to use them in the user selection process, as the corpus collection will be extended to Germany and Switzerland. Further, user names could be used in order to improve filtering and get insights on distributions of language and gender in the corpus (Jaech and Ostendorf, 2015). Last, tweet identifiers can allow for reuse of the corpus (McCreadie et al., 2012) which could also be done with user identifiers.

6. Bibliographical References

Barbaresi, A. (2013). Crawling microblogging services to gather language-classified URLs. Workflow and case study. In *Proceedings of the 51th Annual Meeting of the ACL, Student Research Workshop*, pages 9–15.

Barbaresi, A. (2015a). *Ad hoc and general-purpose web corpus construction*. Ph.D. thesis, ENS Lyon.

Barbaresi, A. (2015b). Collection, Description, and Visualization of the German Reddit Corpus. In *2nd Workshop on Natural Language Processing for Computer-Mediated Communication, GSCL conference*, pages 7–11.

Cheng, Z., Caverlee, J., and Lee, K. (2010). You are where you tweet: a content-based approach to geo-locating Twitter users. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, pages 759–768. ACM.

Ebner, J. (2008). *Duden: Österreichisches Deutsch*. Dudenverlag.

Gonçalves, B., Perra, N., and Vespignani, A. (2011). Modeling users' activity on Twitter networks: Validation of dunbar's number. *PloS one*, 6(8):e22656.

Graham, M., Hale, S. A., and Gaffney, D. (2014). Where in the world are you? Geolocation and language identification in Twitter. *The Professional Geographer*, 66(4):568–578.

Jaech, A. and Ostendorf, M. (2015). What Your Username Says About You. *arXiv preprint arXiv:1507.02045*.

Krishnamurthy, B., Gill, P., and Arlitt, M. (2008). A Few Chirps about Twitter. In *Proceedings of the First Workshop on Online Social Networks*, pages 19–24. ACM.

Kulshrestha, J., Kooti, F., Nikravesh, A., and Gummadi, P. K. (2012). Geographic Dissection of the Twitter Network. In *Proceedings of the 6th International AAAI Conference on Weblogs and Social Media (ICWSM)*, pages 202–209.

Kumar, S., Morstatter, F., and Liu, H. (2014). *Twitter Data Analytics*. Springer.

Leetaru, K., Wang, S., Cao, G., Padmanabhan, A., and Shook, E. (2013). Mapping the global Twitter heartbeat: The geography of Twitter. *First Monday*, 18(5).

Ljubešic, N., Fišer, D., and Erjavec, T. (2014). TweetCaT: a Tool for Building Twitter Corpora of Smaller Languages. *Proceedings of LREC*, pages 2279–2283.

Lui, M. and Baldwin, T. (2014). Accurate Language Identification of Twitter Messages. In *Proceedings of the 5th Workshop on Language Analysis for Social Media (LASM)@ EACL*, pages 17–25.

McCreadie, R., Soboroff, I., Lin, J., Macdonald, C., Ounis, I., and McCullough, D. (2012). On Building a Reusable Twitter Corpus. In *Proceedings of the 35th international ACM SIGIR conference on Research and Development in Information Retrieval*, pages 1113–1114. ACM.

Morstatter, F., Pfeffer, J., Liu, H., and Carley, K. M. (2013). Is the Sample Good Enough? Comparing Data from Twitter's Streaming API with Twitter's Firehose. In *Proceedings of ICWSM*.

Olston, C. and Najork, M. (2010). Web Crawling. *Foundations and Trends in Information Retrieval*, 4(3):175–246.

Ruiz Tinoco, A. (2013). Twitter como Corpus para Estudios de Geolingüística del Español. *Sophia Linguistica: working papers in linguistics*, (60):147–163.

Scheffler, T., Gontrum, J., Wegel, M., and Wendler, S. (2014). Mapping German Tweets to Geographic Regions. In *Workshop Proceedings of the 12th KONVENS conference*.

Scheffler, T. (2014). A German Twitter Snapshot. In *Proceedings of LREC*, pages 2284–2289.

Stone, B. (2009). Twitter Blog: Location, location, location. https://web.archive.org/web/20090823032127/http://blog.twitter.com/2009/08/location-location-location.html.

Zafar, M. B., Bhattacharya, P., Ganguly, N., Gummadi, K. P., and Ghosh, S. (2015). Sampling content from online social networks: Comparing random vs. expert sampling of the twitter stream. *ACM Transactions on the Web (TWEB)*, 9(3):12.

DRuKoLA - Towards Contrastive German-Romanian Research based on Comparable Corpora

Ruxandra Cosma[1], Dan Cristea[2,3], Marc Kupietz[4], Dan Tufiş[5], Andreas Witt[4,6]

[1]University of Bucharest, Faculty of Foreign Languages
[2] *Alexandru Ioan Cuza* University of Iaşi, Department of Computer Science
[3] Romanian Academy, Institute for Computer Science - Iaşi
[4] Institut für Deutsche Sprache, Mannheim
[5] Institute for Artificial Intelligence *Mihai Drăgănescu*, Bucharest
[6] Heidelberg University, Department of Computational Linguistics

ruxandracosma@gmail.com, dcristea@info.uaic.ro, {kupietz, witt}@ids-mannheim.de, tufis@racai.ro

Abstract

This paper introduces the recently started DRuKoLA-project that aims at providing mechanisms to flexibly draw virtual comparable corpora from the German Reference Corpus DeReKo and the Reference Corpus of Contemporary Romanian Language CoRoLa in order to use these virtual corpora as empirical basis for contrastive linguistic research.

Keywords: Reference Corpora, Comparable Corpora, Contrastive Linguistics

1. Introduction

Corpora have increasingly been used in cross-linguistic research, where, in particular, parallel corpora have been of major importance. The usefulness of parallel resources for cross-linguistic research is obvious, as they provide bi- or multilingual, ideally aligned language data that convey the same meaning, including contextual information, and can thus serve as a basis for establishing equivalence between particular entities across different languages (cf. James 1980, Chesterman 1998). On this account, parallel data have been used as an empirical basis in many contrastive studies so far. Some examples include Altenberg (1999), Hasselgård (2007), Zufferey and Cartoni (2012), Kaczmarska and Rosen (2013), where various phenomena from English and Swedish, English, Swedish and Norwegian, English and French, Polish and Czech, respectively, have been accounted for.

Recently, there has also been growing interest in developing comparable corpora (see Sharoff et al. 2013 and the workshop series Building and Using Comparable Corpora) but so far, no comparable resources are available (at least not for German and Romanian) that would allow us to conduct cross-linguistic investigations drawing on language-specific grammatical and semantic properties. The reasons for the DRuKoLA project, as will be sketched in this paper, is to see if a common building strategy can be used for a pair of reference corpora belonging to languages of two diverse families, if a common view on the management of the two corpora can be used and if the access to them can be organised with a common corpus analysis platform. Moreover, the project will investigate how comparable virtual collections (sub-corpora) can be extracted dynamically from this shared resource and how they can serve as flexible, cost-efficient and high-qualitative empirical bases for answering comparative linguistic research questions.

2. Aims of the DRuKoLA project

The DRuKoLA project[1] that is centered around the German Reference Corpus DeReKo (Kupietz, *et al.* 2010) and the Reference Corpus of Contemporary Romanian Language CoRoLa (Tufiş, *et al.* 2015) has started in January 2016 and is a cooperation between the University of Bucharest, the Institute for the German Language in Mannheim, and research institutes of the Romanian Academy in Bucharest and Iaşi. DRuKoLa is a transdisciplinary project involving corpus linguistics, computational linguistics, applied linguistics and cross-linguistic studies, applied computer science, corpus architecture and finally also research infrastructure development. Within this broad range of areas, DRuKoLA's concrete research objectives are:

1. Construction, provision and harmonization of comparable corpora in the two languages.
2. Development of criteria for building comparable virtual sub-corpora based on DeReKo and CoRoLa, the German and, respectively, the Romanian corpus, based on metadata and other possible text properties.
3. Exploration of language-specific peculiarities of the studied languages and equivalences with respect to different parameters and structures.
4. Corpus-based comparative case studies on a) markers of modality: *haben/a avea* with *zu*-infinitives and supine

[1]DRuKoLA is funded by the Alexander von Humboldt-Foundation as a Research Group Linkage Programme between the University of Bucharest and the Institute for the German Language in Mannheim, with the Institute for Artificial Intelligence *Mihai Drăgănescu* (RACAI, Bucharest) and the Institute of Computer Science (IIT, Iaşi) of the Romanian Academy as associated partners. The acronym combines central goals of the project: corpus development and contrastive linguistic analysis (*Sprachvergleich korpustechnologisch. Deutsch - Rumänisch*).

and b) (abstract) demonstratives in German and Romanian, c) general investigation of distributional semantic and syntagmatic properties of corresponding forms and structures.
5. Development of corpus technology to share the corpus, technical and research results in a common Corpus Analysis platform.
6. Building a structure that can serve as a crystallization point for other national or reference corpora with the long-term goal of building a federated, at least European, reference corpus where each corpus is still physically located at and curated by its responsible institute, but can be dynamically extracted to different comparable corpora.

We should also mention that at least the objectives 2 – 5 are planned to be carried out in parallel and in a cyclic bootstrapping fashion. That means for example that the initial naive definition of the comparable corpora and the analysis and visualization functions of the query software will be iteratively refined based on the results of the linguistic analyses. As a welcome side-effect of this procedure, we expect to acquire a good impression of to what extent the linguistic results vary with different corpus compositions and thereby an impression of reliability and generalizability of the obtained findings.
While research objective (6) is also a long-term goal, we already expect numerous synergy effects within the range of current project. First of all, we are convinced that joining national reference or national corpora virtually, with each institute still being responsible for the curation and extension of its own resources is a much more economical and sustainable approach than building multiple comparable corpora from scratch and maintaining them on a project-basis. Another aspect concerns the development and maintenance of sustainable research software that is currently carried out individually for each reference corpus. A closer collaboration in this field with joint forces has the potential of reducing the investments on infrastructure, that are always difficult in the academic context, to a fraction. In addition to such mostly economical arguments, we are convinced that bringing the (corpus-) linguistic communities of different languages together – currently still too much centered around their philologies – has on its own a large boost potential.

3. The underlying corpus resources

Starting a project like this – situated in very different moments of corpus development and architecture – is a rare opportunity, as on the one hand we are working on and witnessing the construction of the Romanian Contemporary Reference Corpus from its beginnings and, on the other hand, are working with a very advanced German reference corpus, analysis system and technology. The collection of data for German started more than 50 years ago and the exploration of principles and methods of empirical anchoring linguistic studies at the IDS in the beginning of the nineties. The project CoRoLa started only in 2014 as a project of national priority of the Romanian Academy. The corpus is rapidly growing, as it is simultaneously being performed in two different institutes of computer sciences, in Bucharest and in Iaşi.

3.1. DeReKo

The German Reference Corpus DeReKo (Deutsches Referenzkorpus) has been developed at the IDS since its inception in 1964. With more than 25 billion words (Kupietz and Lüngen, 2014), it is the world's largest collection of German texts. In contrast to other reference or national corpora, DeReKo is not designed to be used as a monolithic corpus. Instead, it adopts a primordial-sample design approach (Kupietz *et al.*, 2010), which invites users to create stratified sub-samples (referred to as virtual corpora or virtual collections), custom-tailored to their respective research questions and basic populations. Such an approach effectively allows for maximization of its size, diversity and applicability for different research questions (Kupietz *et al.*, 2014) and is also fundamental for the definition of different virtual comparable corpora in the DRuKoLA-context. DeReKo provides a broad variety of text types with a quantitative focus on newspaper texts and rapidly growing portion of computer mediated communication (cf. Beißwenger *et al.*, 2015; Margaretha and Lüngen, 2014; Schröck and Lüngen, 2015). DeReKo is endowed with rich metadata (Klosa *et al.*, 2012; Kupietz and Keibel, 2009), multiply annotated on the part-of-speech, dependency and constituency levels (Belica *et al.*, 2011) and sufficiently licensed to be queried and analyzed for non-commercial linguistic purposes (QAO-NC license, see Kupietz and Lüngen, 2014).

3.2. CoRoLa

Currently, CoRoLa contains more than 191 million word forms of written text and about 135 hours of transcribed speech (Tufiş *et al.*, 2016). In its first public version, CoRoLa will contain more than 500 million word forms and more than 300 hours of transcribed speech (approximately 3 million words) and it will be IPR (Intellectual Property Rights) cleared. It aims at being representative for the literary language. The corpus covers the following 35 subdomains: *literature, politics, gossip, film, music, economy, health, linguistics, theatre, painting/drawing, law, sport, education, history, religious studies and theology, medicine, technology, chemistry, entertainment, environment, architecture, engineering, pharmacology, art history, administration, enology, pedagogy, philology, juridical sciences, biology, social, mathematics, social events, philosophy, other*[2]. The domains and sub-domains classification is based on the Wikipedia one. The functional styles considered are*: journalistic, scientific, imaginative, memorialistic, administrative*, *juridic* and *other* (see footnote 2). CoRoLa uses similar realisation conventions as the Romanian Balanced Corpus (ROMBAC)[3] (Ion *et al.*, 2012) containing over 44 million tokens from five domains (*news, medical, legal, biographic* and *fiction*). The creators of CoRoLa pay special attention in obtaining the consent of owners before including their texts in the corpus; thus, protocols of collaboration have been signed with a number of publishing houses, editorial offices, and radio channels.
In line with the current diversification of language and

[2]This is a category for all documents that could not be definitely classified into the named categories.
[3]http://www.meta-net.eu/meta-share

speech information available in modern representative corpora, CoRoLa will include a syntactically annotated subcorpus and an oral component. All textual data is morpholexically processed (tokenized, POS-tagged and lemmatized). The current annotations are provided in-line but, in the future, as different layers of linguistic annotations (noun phrases, dependency parses, name entities, semantic relations, discourse structures etc.) will be provided for the same data, a mixed (in-line and standoff) annotation will be used. The Universal Dependency (UD)[4] compliant treebank (targeted: more than 10.000 hand validated sentences) and the oral component have additional annotations (dependency links, respectively speech segmentation at sentence level, pauses, non-lexical sounds, like breath, cough, laugh, sneeze, and partial explicit marking of the accent).
The metadata annotators (many of which are volunteers) work under the guidance of a detailed Annotation Manual. Started two years before the initiation of DRuKoLA, the work till now devoted in building CoRoLa was technically supported by an online platform (developed at IIT-Iaşi), which includes facilities for cleaning formatting, standardising Romanian diacritics, eliminating hyphenation, visualizing statistics about the quantity of texts accumulated and their subdomains, and filling in metadata. However, many clearing phases are still done manually: separating articles from periodicals in different files, removal of headers, page numbers, figures, tables, text fragments in foreign languages, excerpts from other authors, and annotation of footers and endnotes (decided to be left in the texts).

3.3. Harmonization of DeReKo and CoRoLa

Both CoRoLa and DeReKo metadata comply with CMDI (Component MetaData Infrastructure)[5] and/or TEI-P5[6] standards. For the construction of comparable corpora, however, in addition to mainly syntactical interoperability, also semantic interoperability has to be achieved, for example for the metadata categories that are used for the construction of virtual corpora. The general procedure for the harmonization of data categories and value sets will be to define functions that map the original respective data to more coarse-grained taxonomies. Additional harmonization work will also be required on lower levels, e. g. for the integration of CoRoLa into the KorAP corpus query engine, or for the adoption of the GGS query mechanism developed for CoRoLa as an auxiliary search engine to express constraints that would exploit the multi-layered annotation of DeReKo, both mentioned in the following section. The first DRuKoLA workshop[7] is expected to answer many of these questions.

4. Query and analysis software

The software that will be used for conducting the corpus linguistic research within DRuKoLA and for making the project results available to the community is the corpus query- and analysis platform KorAP that has recently been developed at the IDS (Bański *et al.*, 2013; 2014). KorAP is the designated successor of the corpus search and management system COSMAS that was launched in 1994 and in its second incarnation (COSMAS II), is currently used by 39.000 German linguists. Besides KorAP's more performance oriented features, like horizontal scalability with respect to an unbounded corpus size and any number of annotation layers, two are particularly fundamental for DRuKoLA: 1.) its ability to manage corpora that are physically located at different places, in order to comply with typical license restrictions (cf. Kupietz *et al.*, 2014) and 2.) its ability to dynamically create virtual sub-corpora based on text properties and to manage these virtual corpora in a persistent way, to e. g. allow for reusability and reproducibility. Further features that will be required for the rather mono-linguistic research purposes will be integrated from recent and ongoing developments of the project partners, as for example the interactive overview visualizations of corpus compositions (Perkuhn and Kupietz, forthcoming), or the visualisation of query expressions as graphs, allowed by the GGS mechanism (Simionescu, 2012). GGS (Graphical Grammar Studio) is an open-source platform allowing interactive writing of grammars that annotate sequences of XML elements at any levels and which has been recently augmented with a constraint-based search mechanism (Simionescu, forthcoming). Also functionalities specifically required for the contrastive research tasks will first be inventoried and then developed during the project.

5. Corpus based contrastive case studies

Based on recent or current research interests of the participating linguists on definite DPs in Romanian (Cornilescu and Nicolae, 2011a; 2011b), situational use of demonstratives (Cosma and Engelberg, 2014) or particularities of the Romanian verbal supine form (Cornilescu and Cosma, 2013; 2014) the project is primarily sustained – as part of the harmonization process – in the making and adapting analyzing instruments for Romanian. The testing phase of the developed instruments will then serve data-based linguistic research and will help identify linguistic variation and preferences within selected research topics: modality markers *haben-zu* infinitives in German and their equivalent finite and nonfinite forms (*are de* V-*ut*, *are* $V_{infinitive}$, *are să* $V_{subj.}$) in Romanian, demonstratives in different uses and positions, reinforcement patterns of demonstratives through adverbs as in *dieser hier*, *dieses schöne Auto da*, propositional reference of demonstratives (*das*, *asta*) etc. Therefore possible aspects to be syntactically explored include: i) distributional patterns of *haben-zu* infinitives with *haben* as a raising verb, distribution of the equivalent form variants of Romanian *are de*/*a*/ *să* + V; ii) identifying structural and stylistic factors in the use of one of the three equivalent forms of the *haben-zu* infinitive in Romanian; iii) the use of propositional demonstrative *das* and singular and plural form differentiated abstract demonstratives *asta*/*astea* in Romanian, etc. For the exploration and analysis of distributional semantic and syntagmatic properties we will use collocation profiles (Belica *et al.*, 2010; Belica, 2011) as well as word embeddings (Mikolov *et al.*, 2013; Ling *et al.*, 2015).

[4] http://universaldependencies.github.io/docs/
[5] http://www.clarin.eu/content/component-metadata
[6] http://www.tei-c.org/Guidelines/P5/
[7] The workshop takes place in April this year in Bucharest

6. Conclusions

We have presented in this paper a very young German-Romanian project, intended to harmonize methods and tools for building and exploiting corpora in these two languages. The idea of the project is to apply a long-standing tradition in the creation of corpora to a newly-born one. At one pole of this project there is the experience gained by the IDS Mannheim in the creation of DeReKo, the largest German language corpus. Two years before this project was initiated, the work on the Contemporary Romanian Language Corpus was simultaneously started in Bucharest and Iaşi. The experience gathered in this period (find providers of texts and vocal recordings, agree on the metadata being used, design and build an interactive platform that helps to clean the linguistic data and fill-in metadata, and design an access mechanism) will now have to be harmonised with the already running German machine. Whether one common methodology will be applicable to both corpora, comparable conventions will have to be fixed through an updating process. This will not only make possible extremely interesting contrastive studies over the two languages and will produce a very large comparative bilingual corpus (with interesting possible beneficials for the MT technology), but the lessons learned from this enterprise could be extended at the European level, to prepare the stage for a multilingual unification of corpora, methodologically and technologically, with tremendous beneficial effects in the multilingual language research.

7. References

Altenberg, B. (1999). Adverbial connectors in English and Swedish: Semantic and lexical correspondences. In Hasselgård and Oksemell (eds.) *Out of Corpora*. Amsterdam: Rodopi, 249-268.

Bański, P., Bingel, J., Diewald, N. , Frick, E., Hanl, M., Kupietz, M., Pęzik, P., Schnober, C. and Witt, A. (2013). KorAP: the new corpus analysis platform at IDS Mannheim. Presented at the 6th Conference on Language and Technology (LTC-2013), Poznań, Polen, December 2013.

Bański, P., Diewald, N., Hanl, M., Kupietz, M. and Witt, A. (2014). Access Control by Query Rewriting: the Case of KorAP. In: *Proceedings of the 9th conference on the Language Resources and Evaluation Conference* (LREC 2014), European Language Resources Association (ELRA), Reykjavik, Iceland, May 2014. 3817-3822.

Beißwenger, M., Ehrhardt, E., Horbach, A., Lüngen, H., Steffen, D. and Storrer, A. (2015). Adding Value to CMC Corpora: CLARINification and Part-of-speech Annotation of the Dortmund Chat Corpus In: Beißwenger, M. and Zesch, T. (eds.): NLP4CMC 2015. 2nd Workshop on Natural Language Processing for Computer-Mediated Communication / Social Media. Proceedings of the Workshop , September 29, 2015 University of Duisburg-Essen, Campus Essen. S. 12-16 - : German Society for Computational Linguistics & Language Technology (GSCL), 2015.

Belica, C., Keibel, H., Kupietz, M. and Perkuhn, R. (2010). An empiricist's view of the ontology of lexical-semantic relations. In: Storjohann, P. (ed.) *Lexical-Semantic Relations. Theoretical and practical perspectives*. John Benjamins Publishing Company. 115-144.

Belica, C. (2011). Semantische Nähe als Ähnlichkeit von Kookkurrenzprofilen. In: Abel, A., Zanin, R. (eds.): *Korpora in Lehre und Forschung*, S. 155-178. Bozen-Bolzano University Press. Freie Universität Bozen-Bolzano.

Belica, C., Kupietz, M., Witt, A. and Lüngen, H. (2011). The Morphosyntactic Annotation of DeReKo: Interpretation, Opportunities, and Pitfalls. In: Konopka, M., Kubczak, J., Mair, C., Šticha, F., Waßner, U. (eds.): *Grammar and Corpora* 2009. Third international conference. Tübingen: Narr. 451-469.

Chesterman, A. (1998). Contrastive Functional Analysis. Amsterdam/Philadelphia: John Benjamins Publishing Company.

Cornilescu A., Nicolae, A. (2011a). Nominal Peripheries and Phase Structure in the Romanian DP. In: *Revue Roumaine de Linguistique* LVI, 1., 35–68.

Cornilescu, A., Nicolae, A. (2011b). On the Syntax of the Romanian Definite Phrases: Changes in the Patterns of Definiteness Checking. In: Sleeman, P., Perridon H. (eds.): *The Noun Phrase in Romance and Germanic. Structure, Variation and Change*. Amsterdam: John Benjamins. 193−222.

Cornilescu, A., Cosma, R. (2013). Restructuring strategies as means of providing increased referentiality for the internal argument of the de-supine clause. In *Bucharest Working Papers in Linguistics* vol. XV.2., 91-121.

Cornilescu, A., Cosma, R. (2014). On the functional structure of the Romanian de-supine. In: Cosma, R., Engelberg, S., Schlotthauer, S., Stănescu, S., Zifonun, G. (eds.): *Komplexe Argumentstrukturen. Kontrastive Untersuchungen zum Deutschen, Rumänischen und Englischen*. Berlin/München/Boston: de Gruyter. [Konvergenz und Divergenz 3]. 283-335.

Cosma, R., Engelberg, S. (2014). Subjektsätze als alternative Argumentrealisierungen im Deutschen und Rumänischen. Eine kontrastive quantitative Korpusstudie zu Psych-Verben. In: Cosma, R., Engelberg, S., Schlotthauer, S., Stănescu, S., Zifonun, G. (eds.): *Komplexe Argumentstrukturen. Kontrastive Untersuchungen zum Deutschen, Rumänischen und Englischen.* Berlin/München/Boston: de Gruyter. 339- 420.

Ion, R., Irimia, E., Ştefănescu, D. and Tufiş, D. (2012). ROMBAC: The Romanian Balanced Annotated Corpus. In Calzolari, Nicoletta et al. (eds.). *Proceedings of the 8th LREC*. 339-344.

Johansson, S. (1999). Corpora and contrastive studies. In Pietilä, P. and Salo, O.-P. (eds.): *Multiple Languages – Multiple Perspectives*. AFinLA Yearbook 1999 / No. 57, 116-125.

Kaczmarska, E., Rosen, A. (2013). Między znaczeniem leksykalnym a walencją – próba opracowania metody ekstrakcji ekwiwalentów na podstawie korpusu równoległego. *Studia z Filologii Polskiej i Słowiańskiej,* 48: 103–121. Warszawa.

Klosa, A., Kupietz, M., and Lüngen, H. (2012). Zum Nutzen von Korpusauszeichnungen für die Lexikographie. In: Lexicographica 28. Berlin/Boston: de Gruyter, 71–97.

Kupietz, M. and Keibel, H. (2009): The Mannheim German Reference Corpus (DeReKo) as a basis for empirical linguistic research. In: Minegishi, Makoto/Kawaguchi, Yuji

(Eds.): Working Papers in Corpus-based Linguistics and Language Education, No. 3. - Tokyo: Tokyo University of Foreign Studies, 53–59.

Kupietz, M., Belica, C., Keibel, H. and Witt, A. (2010). The German Reference Corpus DeReKo: A primordial sample for linguistic research. In: Calzolari, N. et al. (eds.): *Proceedings of LREC 2010*. 1848-1854.

Kupietz, M., Lüngen, H. (2014). Recent Developments in DeReKo. In: Calzolari, Nicoletta et al. (eds.): *Proceedings of the Ninth International Conference on Language Resources and Evaluation* (LREC'14). Reykjavik: ELRA, 2378-2385.

Kupietz, M., Lüngen, H., Bański, P. and Belica, C. (2014). Maximizing the Potential of Very Large Corpora. In: Kupietz, M., Biber, H., Lüngen, H., Bański, P., Breiteneder, E., Mörth, K., Witt, A., Takhsha, J. (eds.): *Proceedings of the LREC-2014*-Workshop Challenges in the Management of Large Corpora (CMLC2). Reykjavik: ELRA, 1–6.

Ling, W., Dyer, C., Black, A. and Trancoso, I. (2015). Two/Too Simple Adaptations of word2vec for Syntax Problems. In: Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Denver, CO: ACL.

Margaretha, E., Lüngen, H. (2014). Building Linguistic Corpora from Wikipedia Articles and Discussions. In: Beißwenger, M., Oostdijk, N., Storrer, A., van den Heuvel, H. (eds.): *Journal for Language Technology and Computational Linguistics* (JLCL) 29 (2). Special Issue on Building and Annotating Corpora of Computer-mediated Communication: Issues and Challenges at the Interface between Computational and Corpus Linguistics. Regensburg: GSCL, 2014, 59-82.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., Dean, J. (2013): Distributed representations of words and phrases and their compositionality. In *Proceedings of NIPS* (Advances in Neural Information Processing Systems) 2013, 3111–3119.

Perkuhn, R., Kupietz, M. (forthcoming): Visualisierung als erkenntnisleitendes Instrument. In Bubenhofer, N. and Kupietz, M.: *Proceedings of the Herrenhausen-Symposium on Visual Linguistics* 2014.

Schröck, J., Lüngen, H. (2015): Building and Annotating a Corpus of German-Language Newsgroups In: Beißwenger, M., Zesch, T. (ed.): NLP4CMC 2015. 2nd Workshop on Natural Language Processing for Computer-Mediated Communication / Social Media. Proceedings of the Workshop, September 29, 2015 University of Duisburg-Essen, Campus Essen. German Society for Computational Linguistics & Language Technology (GSCL), 2015, 17–22.

Sharoff, S., Rapp, R., Zweigenbaum, P. and Fung, P. (eds.) (2013). Building and Using Comparable Corpora. Springer.

Simionescu, R. (2012): Romanian Deep Noun Phrase Chunking Using Graphical Grammar Studio. In *Proceedings of the Conference "Linguistic Resources and Instruments for Romanian Language - ConsILR-2011"*, Bucharest, "Alexandru Ioan Cuza" University of Iaşi Editing House, 135–143.

Simionescu, R. (forthcoming): Symbolic Mechanisms for Describing Linguistic Constraints. Ph.D. Thesis, "Alexandru Ioan Cuza" University of Iaşi.

Tufiş, D., Barbu Mititelu, V., Irimia, E., Dumitrescu, Ş. D., Boroş, T., Teodorescu, N. H., Cristea, D., Scutelnicu, A., Bolea, C., Moruz, A. and Pistol, L. (2015): CoRoLa Starts Blooming – An Update on the Reference Corpus of Contemporary Romanian Language. In *Proceedings of the 3rd Workshop on Challenges in the Management of Large Corpora* (CMLC-3), 5-10.

Tufiş, D., Barbu Mititelu, V., Irimia, E., Dumitrescu, Ş., D., Boroş, T. (2016): The IPR-cleared Corpus of Contemporary Written and Spoken Romanian Language. In: Calzolari, Nicoletta et al. (eds.): *Proceedings of the Tenth International Conference on Language Resources and Evaluation* (LREC'16), Portoroz, Slovenia.

Wälchli, B. (2007): Advantages and disadvantages of using parallel texts in typological investigations. In: *Sprachtypologie und Universalienforschung* 60:2. 118-134.

Metadata Extraction, Representation and Management within the Bulgarian National Corpus

Svetla Koeva, Ivelina Stoyanova, Maria Todorova, Svetozara Leseva, Tsvetana Dimitrova
Department of Computational Linguistics,
Institute for Bulgarian Language, Bulgarian Academy of Sciences
{svetla,iva,maria,zarka,cvetana}@dcl.bas.bg

Abstract

This paper presents the extraction, representation and management of metadata in the Bulgarian National Corpus. We briefly present the current state of the Corpus and the general principles on which its development lies: uniformity, diversity of text samples, automatic compilation, extensive metadata, multi-layered linguistic annotation. The relevant information for the texts in the Corpus is stored into different types of metadata categories: administrative, editorial, structural, descriptive, classificational, analytical, and statistical metadata. The structure and the design of the Bulgarian National Corpus is flexible and can incorporate new metadata categories and values.

Further, we discuss some of the automatic procedures for extraction of metadata applied in the compilation of the Bulgarian National Corpus: (i) metatextual techniques – extracting information from the HTML/XML markup of the original files through a combination of automatic and manual procedures; and (ii) textual techniques – applying text analysis and heuristics using a set of language resources. We briefly present the MetadataEditor – a tool for manual metadata editing and verification.

Directions for future work on the extraction, representation and management of metadata include development of more advanced techniques for language processing, domain-specific analysis, and verification procedures.

Keywords: corpus linguistics, Bulgarian National Corpus, metadata extraction, natural language processing

1. Introduction

The following trends are observed with respect to the relationship between creation of corpora and corpora size, balance and representativeness (Koeva et al., 2012):

- Creation of corpora according to a predefined methodology that is considered sufficiently adequate to ensure corpus balance and representativeness, e.g. the Czech National Corpus (Kocek et al., 2000).
- Development of large unbalanced corpora paired with static balanced subcorpora compiled in accordance with a carefully devised structure, e.g. the National Corpus of Polish (Przepiórkowski et al., 2010).
- Fully automatic compilation of large unbalanced corpora that enables the extraction of subcorpora (Pomikálek et al., 2012).

The approach adopted for the compilation of the Bulgarian National Corpus is based on automatic collection and compilation, detailed metadata description, and multi-layered linguistic annotation. In this paper we focus on the extraction, representation and management of corpus metadata aiming at efficient extraction and compilation of subcorpora with different features and for different purposes. First, we briefly present the current state of the Bulgarian National Corpus and the general principles on which its development lies. Then we discuss the uniform approach to the management of corpus data based on the structure of the metadata. Further, we present the automatic procedures for the extraction of metadata using language technologies, such as keyword extraction, text categorisation, etc.

2. The Bulgarian National Corpus

The Bulgarian National Corpus (BulNC)[1] (Koeva et al., 2010; Koeva et al., 2012) is a large dynamic corpus of Bulgarian consisting of approximately 1.2 billion words distributed in more than 240,000 text documents. The corpus reflects the state of the Bulgarian language from the middle of 20th century until the present. The BulNC also includes parallel corpora of 48 languages of various size, the largest being those for English, Romanian, Greek, and Polish.

The approach we adopt for the BulNC is based on two assumptions: that larger corpora are better suited to linguistic analysis irrespective of the particular task; and that larger corpora, if properly documented and annotated, may also serve as a reliable source from which smaller, uniformly processed, different-sized balanced subcorpora can be extracted, thus eliminating the need for ad-hoc building of standalone fixed-structure corpora (Koeva et al., 2012).

The need for high-quality monolingual and multilingual corpora further necessitates the adjustment of corpus design principles in order to ensure a uniform treatment of monolingual and multilingual corpora, with all texts being documented, processed and accessed within a common framework. The BulNC is developed upon the following principles:

1. Uniform management of multilingual content with respect to compilation, documentation, annotation, processing, and access.
2. No maximum size. The BulNC is a dynamic corpus and new texts are constantly added.
3. Maximum diversity of samples with regard to their form (written, speech), type (style, domain, genre),

[1] http://dcl.bas.bg/bulnc/

lexical coverage (general and specialised lexis), language (multilingual part contains only texts parallel to Bulgarian – original texts or translations).

4. Predominantly automatic collection of corpus samples by means of web crawling based on preliminary manual and automatic web mining, including automatic preprocessing, conversion from html/xml into plain text format, boilerplate removal, elimination of duplicate texts.

5. Corpus structure is managed through a detailed metadata system organised in a classification of categories. The detailed metadata description allows for easy compilation of general, domain- and purpose-specific subcorpora with a fixed structure or predefined features.

6. The metadata are obtained predominantly automatically (due to the size of the corpus).

7. Extensive linguistic annotation is performed by means of dedicated tools (for Bulgarian and English), uniformly represented for all languages and covering different linguistic levels.

3. Uniform Management of the BulNC

In the framework of the BulNC text samples are represented uniformly regardless of their source, size, structure, language or other features. A BulNC text sample consists of two components: text and metadata description, which are stored separately.
The text is stored in the following formats separately:

(a) original as on the source webpage – only used for a limited number of texts, in particular for texts obtained from PDF because their automatic processing is problematic (due to missing information about text components such as figure and table captions, headers and footers, etc.) and new methods for text extraction are still being implemented and tested;

(b) plain text format – this is the main raw format and all texts are stored in plain text;

(c) monolingual annotation – available so far for Bulgarian and English, annotated texts are stored in the widely used vertical format: each token (a word, a constituent of a multiword expression or a punctuation mark) is on a new line with the associated annotations as token type, lemma, POS tag and morphological features in tab-separated fields;

(d) other formats for more complex annotations: aligned parallel texts at sentence level (CSV format), aligned parallel texts at sentence and clause level (XML format), semantically annotated texts (XML format), texts with annotated MWEs (XML format), corpus with extracted citations (JSON format).

Standards for metadata description of linguistic data are provided by the TEI[2], Simple Dublin Core schema[3] and the ISLE Metadata Initiative[4]. While we do not comply with any single standard, our metadata description adopts most of the categories of the above standards, extending them further with a more detailed description (see section 4.).

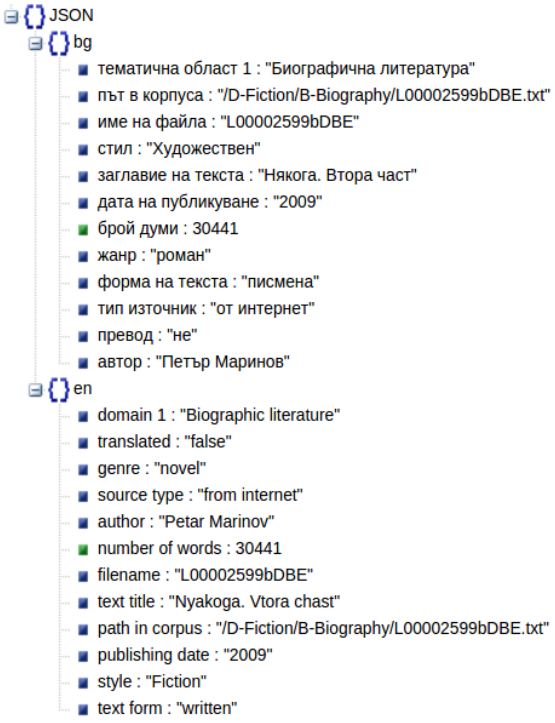

Figure 1: Metadata description in Bulgarian and in English (translated or transliterated and simplified with reduced number of metadata fields)

Metadata are stored separately from the text and are represented in JSON format (see Figure 1). Each text sample has a metadata description file attached which contains the record in Bulgarian, and the description is either translated in English, or transliterated.
We agree with the claim that without metadata, corpus linguistics, being an empirical science, would be virtually impossible (Burnard, 2005). Moreover, we consider metadata as an instrument for effective corpus management. Metadata define the way the samples are organised in the corpus and thus, they are used for identification, management, and retrieval of data (Atkins et al., 1991).
We represent the metadata scheme as an acyclic directed graph (Figure 2) where the nodes are associated with metadata values and the arcs with relations between the nodes expressing metadata categories, such as *style, domain*, and *genre*, etc. For some metadata categories, for instance *style*, the metadata values are predefined; for others, such as *author*, the values are an open set. The representation is simplified, e.g. authorship of the text is recorded only once for all parallel samples in different languages. As a further advantage, graph representation allows flexible extension with new categories and shows where merging or splitting categories is permissible. For example, it is possible to merge the metadata with a database of books' descriptions allowing us to automatically assign publishing dates or ob-

[2]http://www.tei-c.org/release/doc/tei-p5-doc/en/html/
[3]http://dublincore.org/documents/dces/
[4]http://www.mpi.nl/ISLE/overview/overview_frame.html

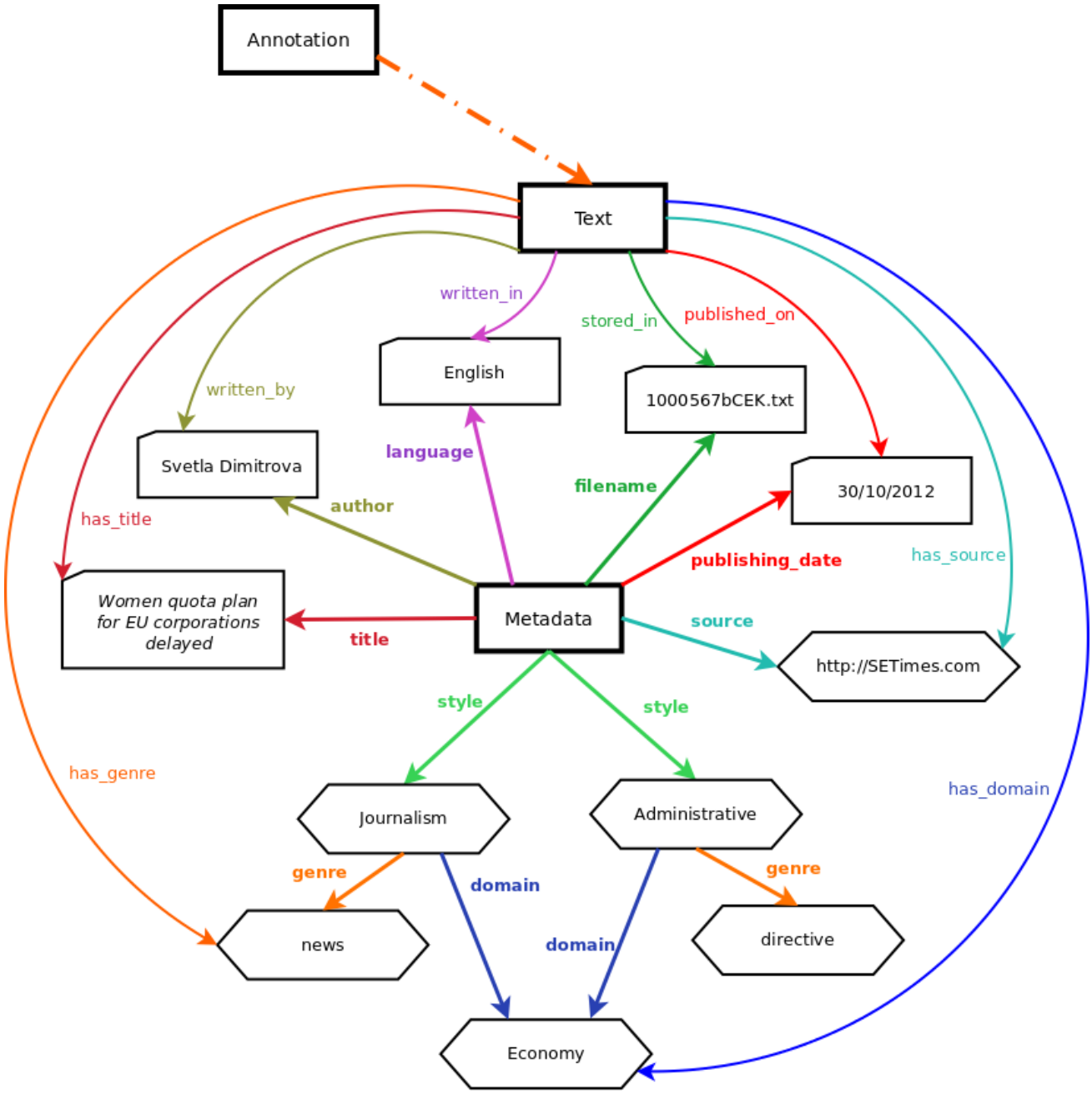

Figure 2: Metadata graph

tain translations of the title in different languages. Different 'graph mining' algorithms – common subgraph, shortest paths, minimum spanning trees, connectedness, etc. can be used when extracting subcorpora of different types.
Through this type of representation we offer the mapping of samples to an interlingual metadata structure. The representation provides three main advantages:

(i) uniformity across languages: metadata categories are used as shared representation by parallel samples in different languages;

(ii) flexibility across different purposes: metadata are not particular to any purpose or application, but can be adapted to new tasks or applications as needed;

(iii) broad coverage: metadata can accommodate a broad range of categories and their values.

The graph representation allows the organisation of metadata in different structures. The metadata are as detailed as possible in order to ensure easy text classification, corpus restructuring and evaluation, derivation of subcorpora based on a set of criteria (e.g., year of publication, domain).

4. Metadata Description

The texts in the BulNC are linked to the relevant values of the metadata categories. Metadata describe the text samples in the corpus and are external (independent of the text) or internal to the text (based on properties of the text). The classification suggested by (Burnard, 2005) is adopted and modified for the description of the metadata:

1. **Administrative metadata** (external) – information about the corpus samples, such as availability, revision status, etc. The administrative metadata used for the description of the BulNC samples are: (i) ***filename***[5]; (ii) *source* (i.e., internet, publishing house, author, etc.); (iii) web address in case the file was downloaded from the internet; (iv) the *date* when the file was added to the corpus; v) access to the file – information where the file is stored.

2. **Editorial metadata** (external) – information about texts in relation to their original source (overlapping, missing parts, etc.) and about the normalisation and editions of samples, if any. The Editorial metadata in the BulNC are: (i) *edited version* (if the text is edited or not); and (ii) *normalisation version* (if the text is normalised or not).

3. **Structural metadata** (external) – information about the relation between the sample and its original source. The Structural metadata in the BulNC are: (i) *number of original texts in the sample*; (ii) *overlapping of the text with original sample* (i.e., exact match, paragraph, random excerption, etc.).

4. **Descriptive metadata** (external) – information about the text, such as: (i) author's name; (ii) author's information (age, sex, nationality, native language); (iii) *title of the text*; (iv) *creation date*; (v) date of publication; (vi) name of publisher; (vii) place of publishing; (viii) text edition (first edition, second edition); (ix) ***language***; (x) ***parallel text*** (yes or no); (xi) *text origin* (original, translation); (xii) name of translator; (xiii) translator's information (age, sex, nationality, native language); (xiv) title of the original text; (xv) ownership of the text; (xvi) notes (any additional information).

5. **Classification metadata** (external and internal) – information used for the basic classification of texts in the BulNC. These include: (i) ***text form*** (written text, speech); (ii) *media* (text, transcription, audio, image, video); (iii) **style**; (iv) text genre; (v) domain.

6. **Analytical metadata** (internal) – various levels of annotation.

7. **Statistical metadata** (internal) (Koeva et al., 2012) – quantitative data for samples: number of tokens, words, lemmas, sentences, etc. The analytical metadata in the BulNC are (i) *number of tokens*; (ii) *number of words*; (iii) *number of multiword expressions*; (iv) *number of sentences*; (v) *number of clauses*; (vi) *number of phrases by type*; (vii) *number of terms*; (viii) *number of named entities by type*.

The following types of metadata categories are observed:

- Based on the number of possible values – open set of values for a given category (e.g., author's name), or predetermined fixed set of values for a given category (e.g., thematic domain);
- Based on optionality of the category in describing the samples – mandatory (e.g., style, text form) or optional value (e.g., notes);
- Based on the number of assigned values – exactly one value (e.g., title of the text) or multiple values (e.g., domain).

The structure and design of the BulNC is flexible and it can incorporate new metadata categories and values. Some of the categories enumerated above are hard to be obtained automatically. Nevertheless, they are included in the metadata schema as a subset of corpora samples was described manually.

5. Automatic Extraction of Metadata

The metadata need to be as detailed as possible in order to ensure easy text classification, corpus evaluation, derivation of subcorpora based on a set of criteria (e.g., publishing year, domain), etc.
The main techniques for automatic extraction of metadata are: (i) metatextual procedures, which consist in extraction of information from the HTML/XML markup of the original files through a combination of automatic and manual techniques with increasing application of the former; and (ii) textual procedures, which consist in application of text analysis and heuristics using a set of language resources.
The following metadata are extracted automatically:

- Editorial and descriptive data.

  The HTML markup of the original files is processed and using a set of patterns, relevant elements are identified and information is extracted. HTML pages usually contain editorial and descriptive information such as author, title, publishing date, specifically marked in the source HTML page.

- Classification information – these include style of the text, register, domain, genre, and result from text analysis.

  In some cases the HTML source may contain classificatory labels according to an adopted domain and/or genre classification on the source webpage, e.g. texts on a news website can be classified into editorials and articles of various domains – Economy, Sport, etc. We identify this information through manual or automatic mining preceding the crawling of the sources, and extract it whenever possible.

- Statistical information – these are derived from processing the text, and include number of words, structure of the text, keywords, etc.

6. Language Technologies for Automatic Extraction and Verification of Metadata

In the early stages of development of BulNC many texts were gathered manually from various webpages. Manual collection of texts is also applied for text categories which

[5]Mandatory categories are marked in bold, categories with exactly one value in italics, and categories with a fixed set of values are underlined.

cannot be found from a single source – e. g. parallel fiction texts.
The automatic collection of corpora was preferred for collecting large amounts of parallel texts and for that purpose a crawler tool was designed. It is adjusted and optimised for each available source (e.g., http://setimes.com/ or http://eur-lex.europa.eu).
The crawler starts at the initial page of the respective archive of documents and recursively harvests the links until the pages containing the documents are reached. Web structure mining is employed in the crawler design to reduce the number of visited links and to improve efficiency.
The development of efficient methods for automatic compilation and verification of the metadata description is essential for ensuring the high quality of the resources and supports their flexibility and adaptability for various research purposes.
Language technologies can be further applied to improve metadata description. A set of modules are used to process the texts in order to derive metadata directly or to extract additional data that after being analysed, lead to extending the metadata or to validating them.
For the purpose of metadata extraction and verification we employ the following resources:

- **MWEDict**: Dictionary of Multiword expressions (MWEs) and Named entities (NEs) – containing 27,744 nominal MWEs, out of which 18,962 are NEs such as geographical names, events, etc. (Stoyanova and Todorova, 2014).
- **DomMapDict**: Dictionary mapping keywords to a domain – containing 3,581 words indicative of the domain (e.g., Economy, Political, Botanics, etc.).
- **DomSpecDict**: Dictionary of 23,203 domain-specific single words and MWEs from various domains, derived from Wikipedia, specialised dictionaries, etc.

6.1. Keyword Extraction

The implementation of keyword extraction uses frequency analysis of simple words and N-grams. The text is initially processed using the Bulgarian Language Processing Chain (Koeva and Genov, 2011). Already known MWEs and NEs (found in the MWEDict dictionary) are identified.
For the purposes of metadata extraction and validation a simple word or a MWE is considered to be a keyword in a text if it appears with a relatively high frequency in the text and it is essential in characterising the text. It can either be a word that has already been labelled as characteristic for the text, e.g. it appears in the title or in any other metadata (subtitle, category, etc.); or a word that is generally associated with a certain domain and thus, is considered to be highly informative as a keyword, e.g. if it is the name of a text category in the classification.
Keyword extraction can be applied on both raw (token-based) and annotated text (lemma-based). Although it is more reliable when bigger data are used, it is still possible to apply it on a single text.
The module for keyword extraction applied on a single text follows these steps:

1. Checks frequency of words from the title and other preset metadata and filters out words with frequency below threshold N_1;
2. Checks frequency of domain-specific words (from the dictionary DomMapDict) and filters out words with frequency below threshold N_2;
3. Identifies lemmas with frequency above certain threshold N_3;
4. Identifies N-grams (currently only bigrams) with frequency above threshold N_3.

The thresholds N_1 and N_2 are currently set to lower values ($N_1 = N_2 = 3$) than N_3 ($N_3 = 5$) since the first two groups are more likely to represent an informative keyword as they have been selected manually either when publishing the text or when constructing the dictionary.
Other approaches can work on the whole corpus and apply statistical measures for identifying keywords, such as tf/idf. Keywords extraction is essential in order to identify or validate categorial information such as style, genre, and domain.

6.2. Text Categorisation

The style of the text, or the general text type, is usually known from the source (e.g., news are collected from news websites, administrative from government websites, etc.). Whenever possible, additional classificational information is extracted from the HTML markup.
The genre of the text is identified in one of the following ways:

(i) from the source if the text is labelled as belonging to a certain genre on the website (e.g., editorial or news);

(ii) from the title if the genre is present in it (e.g., in administrative texts, ***Decision*** *of the European Commission 2001/711/EC of 29 June 2001*);

or

(iii) via textual analysis – length of text (e.g., to distinguish between a novel and a short story), structure of text (e.g., poems), etc.

Text categorisation with respect to domain is predominantly based on extracted keywords and involves the following steps:

1. Keywords from the dictionary DomMapDict directly map the text to the corresponding domain.
2. Each text can be assigned more than one domain, which are ranked; the frequency of identified keywords gives weight to assigned domains. E.g., if the word *political* occurs 7 times in a text, and the word *economy* 15 times, the ranked list of assigned domains is *Economy, Politics*.
3. Other keywords from the text identified on the basis of frequency, if they are domain-specific terms (found in DomSpecDict), can also be used to give weight to predicted domains and ensure more reliable ranking.

6.3. Tool for Manual Verification

MetadataEditor 1.0 (Figure 3) is a tool for manual metadata editing and verification. It was developed in Java 7 and is compatible with different operating systems. The program can work with the metadata scheme of the BulNC, or can use a different scheme and corpus classification in JSON format.

The program loads the text sample as a pair of files – a metadata file and a text file. The user can edit both the metadata and the text. Some metadata fields have a fixed number of possible values (e.g., style, domain, genre) and are provided as a drop-down list, while others are free text (e.g., title, author, keywords) presented as a textbox. In addition, some fields can contain a list of values rather than a single value (e.g. a text can belong both to domains Medicine and Biology). These properties and the lists of values (where applicable) are provided in a separate file which describes fully the metadata scheme (BulNC or user defined).

The tool provides a multilingual mode in which one may check, supplement and/or synchronise the metadata of parallel documents in two or more languages. It has been tested on the pair Bulgarian–English mainly for practical reasons as currently the greatest amount of parallel texts and the most variety of preprocessing tools and resources (dictionaries, gazetteers, etc.) at our disposal are for this language pair. Extension of the metadata to edit and/or validate is possible for any (other) set of languages.

Once the primary language metadata are loaded (in our case Bulgarian), the pertaining metadata for the second language (in our case English) are loaded as well. Any changes in the primary language metadata need to be automatically synchronised with the second language metadata record through a Synchronise functionality, selected from the menu. In case changes involve metadata categories with close-set values, the values in the second language are translated using a preliminary compiled list. Open-set values are transliterated automatically. One of the directions of improvement is to consider which of the open-set metadata values should better be translated or otherwise rendered in the secondary language.

Some additional data-processing functionalities are provided, such as a spell-checker to be applied on the text (to identify texts of poor quality) and keyword extractor. The tool can easily be extended to include new functionalities.

The automatically extracted metadata subset of 9,014 samples of the BulNC amounting to 168 million words was checked and validated manually using MetadataEditor 1.0. As a result, 32.3% of metadata records have been amended, where 7% of all metadata fields were supplied with new information and another 5.7% of metadata fields were edited. The most frequently edited metadata fields are *publisher*, *periodical* and *issue*, *place of publishing*, and *keywords*.

7. Conclusion and Future Work

Our approach emphasises on the extensive metadata of very large (predominantly) automatically collected monolingual and multilingual parts of BulNC. Several types of access to the corpus are provided:

(i) download (limited);

(ii) web search interface;

(iii) subcorpora selection;

(iv) frequency lists derived from the whole corpus or a given subcorpus.

Several directions for improving both the process of metadata compilation and the editing tool emerged from the validation procedure.

1. *Enhancing keyword extraction*. Currently keywords are extracted based on frequency. The approach may be supplemented and perfected in several directions:

   (i) Considering discovering synonymy and relatedness between keywords by means of lexicons of words and semantic relations (e.g. ones extracted from WordNet) or by calculating similarity measures from WordNet or another (conceptual) resource.

   (ii) Providing relations between standard names and abbreviations (*European Union* and *EU*), alternative names, transliterated versions, etc.

   (iii) Incorporate a stemming procedure to discover 'families' of keywords belonging to different parts of speech and/or different derivational models.

   An improved procedure for keyword extraction will impact the reliable assignment of documents to (a) particular domain(s) and will be helpful in any task based on similarity and relatedness of documents.

2. *Adopting a differentiated treatment of synchronisation of changes in open-set values made in the Bulgarian part of the metadata to a second language metadata (English or other languages)*. The rendition of these values depends on the type and original language of the respective text.

   (i) For Bulgarian names (authors, translators, and possibly publishing houses, journals, newspapers, etc.) the appropriate treatment is transliteration. Transliteration is made automatically through a built-in function of the tool.

   (ii) Foreign names should be rendered with their standardised representation in the language of the metadata (if there is one), or as spelled in their original language. This information may be supplied if it is available in the document or its metadata description (if there is one, e.g. in the source html file).

   (iii) Bulgarian titles may be additionally supplied both with the transliteration and with a standardised translation in case the text has been translated.

   (iv) In the general case, foreign titles should be rendered with their standardised representation in the language of the metadata (if there is one), or as in the original language if this information is extractable from the document.

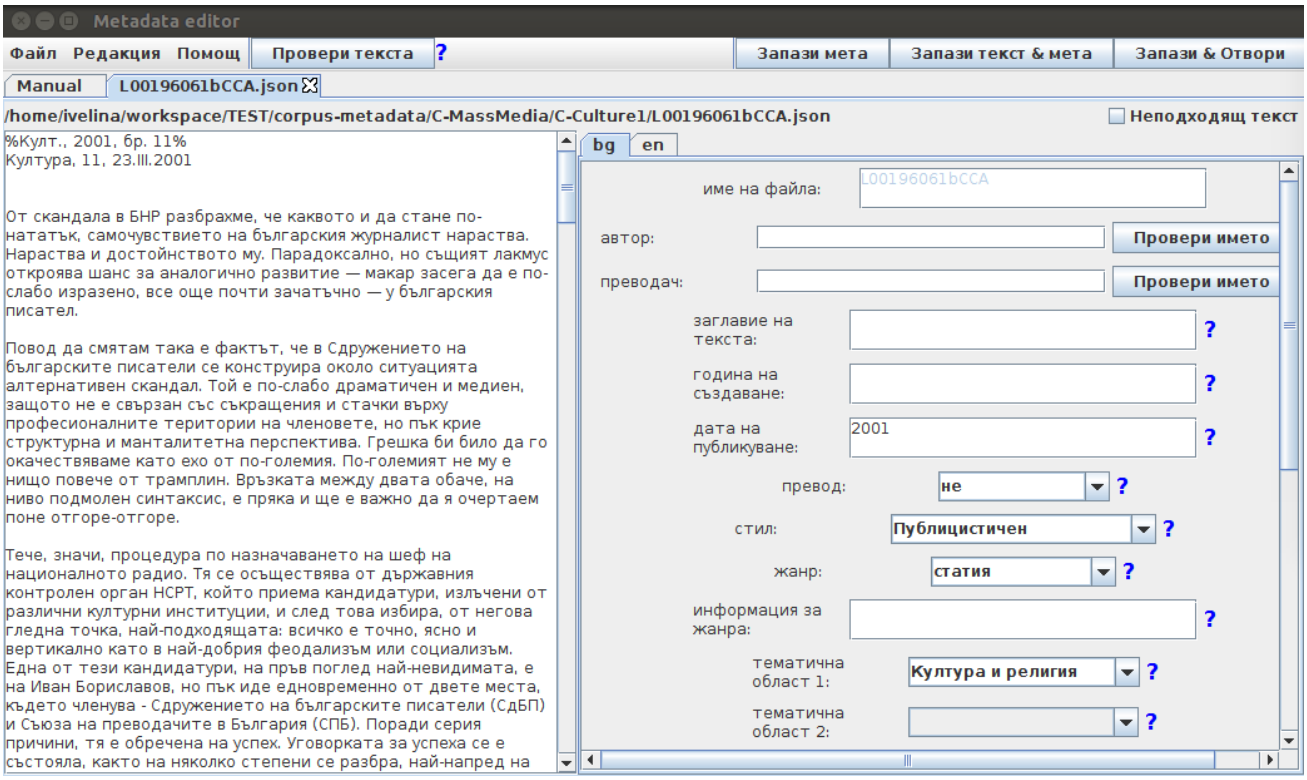

Figure 3: MetadataEditor 1.0

We are going to study how the automatic metadata extraction can be improved. One possible way is to improve the linguistic resources involved (dictionaries) and possibly to introduce new types of resources, for example WordNet. Other direction is to involve new algorithms, ranging from domain-neutral to domain-specific texts.

8. References

Atkins, S., Clear, J. H., and Ostler, N. (1991). Corpus design criteria. *Journal of Literary and Linguistic Computing*, pages 1–16.

Burnard, L., (2005). *Developing Linguistic Corpora: a Guide to Good Practice*, chapter Metadata for Corpus Work. Oxford: Oxbow Books.

Kocek, J., Kopřivová, M., and Schmiedtová, V. (2000). The Czech National Corpus. In *Proceedings of EURALEX 2000*, pages 127–132.

Koeva, S. and Genov, A. (2011). Bulgarian language processing chain. In *Proceeding to The Integration of multilingual resources and tools in Web applications Workshop in conjunction with GSCL 2011*. University of Hamburg.

Koeva, S., Blagoeva, D., and Kolkovska, S. (2010). Bulgarian National Corpus Project. In N. Calzolari, et al., editors, *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC 2010)*, pages 3678–3684.

Koeva, S., Stoyanova, I., Leseva, S., Dekova, R., Dimitrova, T., and Tarpomanova, E. (2012). The Bulgarian National Corpus: Theory and Practice in Corpus Design. *Journal of Language Modelling*, 0(1):65–110.

Pomikálek, J., Jakubíček, M., and Rychlý, P. (2012). Building a 70 billion word corpus of English from ClueWeb. In Nicoletta Calzolari (Conference Chair), et al., editors, *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12), 2012, May, 23-25, Istanbul, Turkey, ELRA*.

Przepiórkowski, A., Łaziński, M., Górski, R. L., and Lewandowska-Tomaszczyk, B. (2010). Recent Developments in the National Corpus of Polish.

Stoyanova, I. and Todorova, M. (2014). Razrabotvane na rechnitsi na sastavnite leksikalni edinitsi v balgarskiya ezik za tselite na kompyutarnata lingvistika. In *Ezikovi resursi i tehnologii za balgarski ezik*, pages 185–202. Academic Press Prof. Marin Drinov.