tranScriptorium

D4.3.2: Description of linguistic resources for HTR

INL

Distribution: Public

tranScriptorium
ICT Project 600707 Deliverable D4.3.2

Project funded by the European Community
under the Seventh Framework Programme for
Research and Technological Development.

Project ref no.	ICT-600707
Project acronym	**tranScriptorium**
Project full title	**tranScriptorium**
Instrument	STREP
Thematic Priority	ICT-2011.8.2 ICT for access to cultural resources
Start date / duration	01 January 2013 / 36 Months

Distribution	Public
Contractual date of delivery	December 31, 2015
Actual date of delivery	December 31, 2015
Date of last update	January 18, 2016
Deliverable number	D4.3.2
Deliverable title	Description of linguistic resources for HTR
Type	Report
Status & version	Draft
Number of pages	24
Contributing WP(s)	WP Number
WP / Task responsible	Consortium members
Other contributors	
Internal reviewer	Joan Andreu Sánchez
Author(s)	INL
EC project officer	Officer name
Keywords	

The partners in **tranScriptorium** are:

Universitat Politècnica de València - UPVLC (Spain)
University of Innsbruck - UIBK (Austria)
National Center for Scientific Research "Demokritos" - NCSR (Greece)
University College London - UCL (UK)
Institute for Dutch Lexicology - INL (Netherlands)
University London Computer Centre - ULCC (UK)

For copies of reports, updates on project activities and other **tranScriptorium** related information, contact:

The **tranScriptorium** Project Co-ordinator
Joan Andreu Sánchez, Universitat Politècnica de València
Camí de Vera s/n. 46022 València, Spain
jandreu@dsic.upv.es
Phone (34) 96 387 7358 - (34) 699 348 523

Copies of reports and other material can also be accessed via the project's homepage: http://www.transcriptorium.eu/

Contents

Executive summary

The aim of work package WP4 of **tranScriptorium** (tS) is to provide and present a general approach and workflow to acquisition and integration of linguistic resources, to collect linguistic resources relevant to the manuscript collections tackled in the project and to build *an optimized language modeling for HTR* in order to improving HTR performance. The contribution of the developments of this WP for improving HTR results will be explicitly evaluated in tasks T3.5 and T5.5.

This document describes approach to development and deployment of linguistic resources, provides a concise description of the individual resources, and contains an evaluation of the contribution of the developed resources to HTR quality and text accessibility.

We furthermore describe our approach to the deployment of language resources in the *language resource server*.

The companion document (D4.2.2) describes the toolbox for language modeling (deliverable D4.2.1), tools for workflow and deployment of linguistic resources for HTR.

1 Introduction: Language resources for Handwritten Text Recognition

An indispensable component of state-of-the-art HTR is language modeling, which is necessary to guide the decoding process by ranking and constraining the possible word sequence hypotheses. Language modeling has proven extremely successful in improving results of Automatic Speech Recognition, which is a very similar task from the technical point of view. Highly effective language models in this field have been developed from huge language corpora. Language models are usually constructed from large text corpora which – ideally – are *in-domain*, linguistically close to the language of the document collection which is being processed.

However, for HTR of historical documents, obtaining effective models is much less straightforward: models built from the strictly in-domain data are generally unsatisfactory because not enough data can be obtained to avoid overfitting, and in order to exploit the larger pool of out-domain data one has to surmount two difficulties: (1) indiscriminate use of *out-of-domain* data may not benefit, in fact even deteriorate system performance and (2) the use of the complete out-domain data for training may increase the complexity of the system, making the decoding process almost untractable.

The above-mentioned issues are typically dealt with by using *domain adaptation* techniques, which aim to leverage the knowledge that can be obtained from the out-of-domain data by tuning it to the in-domain data. The methods we have been developing and using in the project are described in D4.2.2. We need not go into the details here, we need to retain from this that a good set of language resources for HTR crucially depend on the availability of both general-purpose and very specific material close to the collection we are dealing with.

The following types of language resource are useful either directly in HTR or in postprocessing or transcription or post-editing.

1. **Text corpora** provide background material from which language models or lexica can be developed. We will need both smaller, specific corpora and larger general-purpose corpora.
2. **Lexica** for a collection contain words relevant for the collection. They can be deployed as HTR lexica or for lexical suggestions in during transcription or post-editing
3. The **HTR lexicon** contains lexical information in the form that can be used in the recognition process
4. **Language models** are deployed directly in HTR to guide the recognition process.

2 Approach to the deployment of language resources

2.1 Problem statement

As opposed to a static system which is developed once and does not change during deployment, the development of an HTR system, including language resources, is a continuous process. During transcription or post-editing of material, more training materical becomes available for the core HTR process; processing parameters may be optimized for the collection. Similarly, it is not appropriate to deliver a static set of HTR language models and lexica for a certain language, type of documents or even a collection.

We summarize the main issues in obtaining relevant resources for HTR, leading to the conclusion that appropriate resources are a moving target.

1. The HTR training set and the set of available external language resources evolve during transcription of a collection
   (a) By itself, the growing training set calls for retraining HTR and the language modeling component
   (b) When we view HTR and transcription in the context of a collaborative platform like *Transkribus*, we should be able to benefit from transcriptions of similar documents that take place in the platform.
2. The resources must be relevant and adapted to the material being processed. This relevance can be best assessed from the HTR training set.
3. HTR lexica and language models depend on text parameters that may change. In particular:
   (a) The character set accepted by the HTR system may grow or change
   (b) Tokenization rules may be modified as a consequence of HTR system optimization
4. Apart from this, we want to make language resources available for different HTR systems, in such a way that we can compare the merits of the systems that make use of the resources

From the above, we may conclude that the best way to deliver language resources is not just to make the final result (HTR lexica and language models) available, but also

1. The source material from which models and lexica are constructed
2. The tools to construct them

2.2 A server-based collaborative approach

To address the above issues, we decided to opt for a server-based solution, which is in keeping with the general design behind the collaborative approach, where we envisage to enable the HTR community to contribute resources. More in particular, the language resource server

- Stores an evolving set of language resources with relevant metadata like language, period, ...
- Includes the tools necessary to compile the resources to the formats that can be deployed directly in HTR.

An additional benefit of a server-based approach is that in many cases, we may not be allowed to (re)publish text material, whereas statistics obtained from the material, like language models and/or lexica, are far less problematic.

3 The language resource server

The language resource server is a tomcat servlet component with a simple REST-style interface. It stores documents and metadata in a PostgresQL database. For language-modeling operations, it relies on Berkeley LM, SRILM, and HTK. A full technical description will be included in appendix 1 (forthcoming).

3.1 Repository functions

An important function of the LR server is to provide an open repository of language resources, to which users may upload relevant material and provide relevant metadata. Accordingly, the LR server has functions to upload, download, search documents and to add metadata attributes.

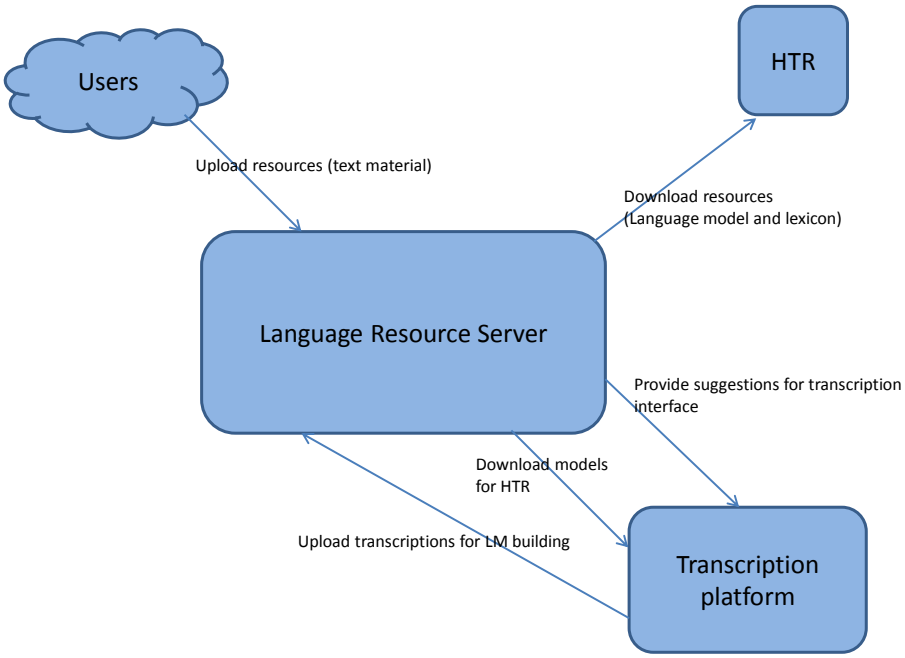

Figure 1: Interactions with the language resource server

3.2 Language model building functions

Construct LM and HTR dictionary from corpus text, parametrized by character set, tokenizer

3.3 Language model deployment functions

- Context-dependent word completion suggestions from Language model or dictionary, to be used in transcription user interfaces
- Statistical functions: Perplexity computation, Out-of-vocabulary rate computation

4 Description of language resources

We provide a general description of the resources here. A more complete listing of the files as they appear in the Language Resource Server will be included in appendix 2.

4.1 Language resources for Dutch

4.1.1 Corpora

Apart from the ground truth data which is used for HTR experiments, we have several data sets which are used for Language model training.

- *The complete Hattem manuscript* (apart from the pages used for HTR training), consisting of 600 pages (68000 tokens).

- *The Dutch artes corpus collected by INL*, consists of a set of artes manuscript transcriptions from various sources. Notably:
  - 21 editions from DBNL [1], which have been processed to separated middle Dutch text from editorial text. This corpus consists of bout 325000 tokens
  - 26 editions from the Middle Dutch Corpus (cf. below), about 387000 tokens
- *The INL Middle Dutch Corpus*, A larger background corpus, to be used as out-of-domain material. Converted to TEI for use in the tranScriptorium project. It consists of about 4M tokens of prose and about 4.5M tokens of rhymed text.
- For the "Resolutions" collection, a corpus of about 1M tokens of related material has been acquired and processed (the "Stellingwerf" corpus).

4.1.2 Lexica

A very extensive lexicon for text normalisation has been developed from the MNW (Dictionary of Middle Dutch).

4.1.3 Language models

For all collections described in the following section, language models have been developed using different ways of combining the different corpora avalailable for Dutch. Results are described in section 6

4.1.4 Varia

A catalogue of relevant abbreviation symbols and their expansions has been constructed.

4.2 Language Resources for German

4.2.1 Corpora

- A Corpus of historical documents from Zettl has been acquired and converted to TEI.
- The core historical corpus from the DTA (Deutsches Textarchiv) has been acquired.
- Specifically for the Reichsgericht collection: We have compiled a small collection of similar transcriptions
- We have prepared and processed the Project Gutenberg [2] German subcorpus for use as out-of-domain background corpus

4.2.2 Lexica

Relevant word lists have been acquired.

1. A frequency list from the DTA (Deutsches Text Archiv) corpus
2. Headword lists from DWDS-Wörterbuch, the first edition of the Grimm dictionary and the Pfeifer etymological dictionary.

[1] `http://www.dbnl.nl`
[2] `http://www.gutenberg.org`

4.3 Language resources for English

4.3.1 Corpora

Apart from two data sets for HTR experimentation (cf. below), we use the remaining corpus of transcribed manuscripts (about 15.000 pages and 5m words) als language model training data. We have also used versions of Bentham's printed works available from the *Online Library of Liberty*[3].

The public part of the ECCO (Eighteenth Century Collections Online) [4], about 70m words, is used as out-of-domain data.

4.3.2 Lexica

Permission has been obtained from Oxford University Press to use the lexicon developed in the IMPACT[5] project, based on the Oxford English Dictionary, as an auxiliary resource for text normalisation. This has been used in processing, but cannot be delivered as a project deliverable.

4.3.3 Language models

Since the Bentham corpus has been used extensively, many different versions of language models of different order (up to 5-grams), have been obtained during the project. They are lissted in appendix 2.

4.4 Language resources for Spanish

4.4.1 Corpora

1. We have obtained permission from the Cervantes Virtual library to use BVMC corpus material. This material has been acquired and consists of
2. We have obtained permission from the University of Alicante to use of the IMPACT-developed historical lexicon for text normalization purposes
3. We have downloaded 390 Spanish books from Project Gutenberg, converted them to TEI, solved encoding issues, discarded English language additions with the help of language identification to obtain a cleaned corpus of about 25 million running words.

4.4.2 Lexica

Permission has been granted by the University of Alicante to use the IMPACT-developed historical lexicon for text normalisation purposes. This resource has been acquired.

4.4.3 Language models and HTR dictionaries

5 Description of relevant document collections

In this section, we describe, per language, the document collections we have used for experiments, the language resources developed for the languages, and the results of relevant HTR experiments.

[3] `http://oll.libertyfund.org/`

[4] `http://www.textcreationpartnership.org/tcp-ecco/`

[5] `http://www.digitisation.eu`

5.1 English language data

5.1.1 Document collections

For english, HTR experiments concentrate on subsets of documents from the Bentham collection, which have been prepared in the tranScriptorium project. This dataset includes manuscripts written by Jeremy Bentham (1748-1832) himself over a period of sixty years, as well as fair copies written by Bentham's secretarial staff. Handwriting in this collection is complex enough to challenge the HTR software: manuscripts written by secretarial staff will provide variety, while Bentham's manuscripts are often complicated by deletions, marginalia, interlineal additions and other features (Gatos, 2014).

5.2 Spanish language data

5.2.1 Document collections

5.2.1.1 Plantas

Historia de las Plantas is a manuscript written by Bernardo Cienfuegos, who was one of the most outstanding Spanish botanists in the 17th century. Historia de las Plantas manuscript is conserved at the Biblioteca Nacional de Espaa (a digital extract of the manuscript is available at Biblioteca Digital Hispnica). Historia de las Plantas attempts to synthesise the botanical knowledge along the history. It is written in Spanish, it is well documented and details of great scientific interest can be found in this book.

The manuscript has 7 volumes with different number of pages each one, but just the first volume is currently being processed. This volume has over 1000 pages, and it includes drawings of plants, some in colour and others in black and white.

Historia de las Plantas contains a large number of Latin names of the species and other non-Latin terms: Greek, Arabic, Hebrew, Portuguese, Valencian, Catalan, French, German, English, Flemish, Polish, Bohemian and Hun. This makes this dataset challenging from the point of view of language modeling.

5.2.1.2 Alcaraz

The "Alcaraz" dataset is a collection that has been recently introduced into the tranScriptorium project. It is a handwritten record of the oral declarations made by Pedro Ruiz de Alcaraz during his Inquisition process (1534-1539). The document is in Spanish of the 16th century and composed of 953 pages written in a style known as "Procesal encadenada". This style is characterized by being quick on-line writing, without consistent blanks between and within words, and with plenty of (rather inconsistent, often improvised) abbreviations.

5.3 German language data

5.3.1 Document collections

5.3.1.1 Reichsgericht

Reichsgericht is a dataset composed of court decisions from the High Court of Germany. For 114 pages of this dataset a perfect GT has been obtained, where the transcription and the detection of the lines (at baseline level) has been carried out automatically.

The 114 pages were used to carry out HTR experiments. They were divided into two blocks, the first one composed of 88 pages was used to train HMM and LM. The second one, composed of 26 pages, was used to test the system.

5.4 Dutch Language data

5.4.1 Manuscripts

We concentrate on a collection of Middle Dutch late medieval manuscripts belonging to the "Artes Liberales" literature, which were mostly written in the cursive style. Most texts belong to the 15^{th} century and belong to the medical domain. These documents are of great interest for the history of science, language and literature alike, and their transcription will be all the more valuable as not many documents of this type have been edited.

The fact that they belong to a single subject domain makes them and interesting test case for domain adaptation techniques.

5.4.1.1 The Hattem manuscript

The *C5 Hattem Manuscript*[6] (15th century, 572 leaves), has been completely transcribed by the WEMAL group[7], which makes it very suitable for experimentation. Apart from a small number of Latin and French documents, most of the texts are in Middle Dutch. The contents are very heterogeneous. There is a prose translation of the Secretum Secretorum (a Latin translation of an Arabic encyclopedia on government, health, astrology and alchemy), and a Dutch treatise on the plague, which is ascribed to the pope. The subject matter of the Dutch language treatises includes phlebotomy, surgery and uroscopy. There is an extensive treatment of herbalism, various recipes for medicinal potions, oils and unctions, alchemist procedures, and a specialised treatise on precious metal alloys.

5.4.1.2 The Leiden Manuscript

This manuscript has been digitized at the Leiden university library as *BPL 3094*. It is dated between 1475 and 1500, and consists of 159 folio's or 318 pages. A subset of 96 pages has been transcribed for HTR experiments.

5.4.1.3 The Meermanno Manuscript

This is catalogued as 10 C 17 at Museum Meermanno, The Hague. Date is around 1470. full manuscript size is 208 folio's or 416 pages. 50 Pages (100 columns) have been transcribed for HTR experimentation.

5.4.1.4 The Resolutions collection

The transcripts of the Resolutions of the States-General consist of 200.000 pages of handwritten text that not only reflect the invention and early development of the new Dutch State, but also are a witness of the daily political activities of the *Hoogmogende Heren*.

A pilot set of 17-th century hand-written resolutions has been selected, consistiong of 40 two-page images. 20 have been transcribed. Training set is 15 pages, test set is 4 pages.

[6] Utrecht university library, MV : C5,
`http://objects.library.uu.nl/reader/index.php?obj=1874-44915&lan=en`

[7] `http://wemal.let.uu.nl/hattem-c5.html`

6 Evaluation of language resources

This section describes the results of HTR experiments conducted during the tranScriptorium project, which are relevant to language model development. Some of these results are still susceptible to change in the coming weeks, as experiments are not always finalized. For each dataset, we try to assess the amount to which inclusion of additional language resources is beneficial.

The following quantities are obviously most relevant to assess HTR accuracy

- Word error rate using "default" language model, i.e. the one obtained by using the training set for language modeling
- Word error rate using the enhanced language model

We must bear in mind that besides first-best transcription quality, the potential of retrieval using string-based word spotting approaches on word lattices is also very important. Taking into account that these methods, without special modifications, are only able to retrieve words that are present in HTR lexicon and language model, the following quantities are of interest as optistic bounds for recall of string-based word spotting.

- *Lattice word error rate*: by oracle choose best path through word graph, compare this to truth (lower bound for WER improvement that can be achieved by language modeling)
- *Lattice word recall*: which portion of ground truth words are (at all) hypothesized by the system

Of course, an evaluation of a language resource cannot be carried out independently from the technique which has been used to deploy it - there is alwas room for improvement by refining the deployment methods.

One further remark: most of the results reported below have not been obtained with the most sophisticated optimized instance of the HTR system. This accounts for some differences in reported WER in this report and the reports from work package 3.

6.1 Dutch

Note, again, that we are not using optimal HTR settings but a system default. These results do not represent the optimal systems.

We have used, in these experiments, linear interpolations of language models build from the following collections

1. The part of the HATTEM manuscript that is not used ofr HTR training or testing
2. The corpus of 'DBNL' liberal arts material
3. The corpus of liberal arts material from the INL middle Dutch corpus
4. The corpus of prose texts from the INL middle Dutch corpus
5. The corpus of rhymed texts from the INL middle Dutch corpus

Dataset	WER def.	WER enh.	CER enh.	Lattice WER def.	Lattice WER enh.
Hattem	34.36	29.87	12.86	25.24	13.36
Leiden	49.6	43.66	21.54		
Meermanno	42.2	36.89	14.96		
Resolutions	48	40.4	13.01		18.42

Results here lead to the conclusion that we may not yet have found the optimal way to deploy our language resources for Middle Dutch yet in first-best transcription, but they are already quite useful, especially in improving the recall of string-based keyword spotting, as can be seen from the good Lattice WER rates.

6.2 English

Table 1: Results of language modeling techniques on the Bentham competition set

Method	WER (case sensitive)%	WER (case insens.)
In-domain data + Unigram LM	24.28	23.5
In-domain data + Bigram LM	21.96	21.1
Adapted Bigram LM	17.33	16.5
Adapted Trigram LM	16.53	15.7
Adapted 4-gram LM	16.52	15.6
Adapted 5-gram LM	16.94	16.1

Table 1 gives a brief summary of the work on English, which has been reported on elsewhere (cf. D422). One should compare this to the best results reported in work package 3, where an improved tokenization approach enables better recognition of punctuation characters, with a resulting WER of 18%. An enriched bigram language model with this improved tokenization reduces this to around 15%. A trigram model could presumably improve further on this, but this experiment is still pending.

6.3 German

In this basic experiment, we have used an interpolation of

1. A model obtained from the HTR training set
2. A model obtained from the set of OCR-ed transcriptions
3. A model obtained from a selected sample of the Gutenberg corpus. The sample selection method used was very simple: we used the 50 books with lowest perplexity with respect to model 2.

Dataset	WER def.	WER enh.	CER def.	CER enh.	OOV def.	OOV enh.
Reichsgericht	33.8	26.8	14.49	11.87	10.26	3.96

6.4 Spanish

Experiments with the Plantas and Alcaraz collections are still pending. Both collections consist of very specific material, and use transcription methods which make it difficult to involve external language modeling material.

Appendix A: language resource server

Resource repository

The resource repository is a simple file database, supporting assignment of metadata to resource items. As various implementations, using different storage backends, seem possible, we defined a simple function interface that should be satisfied by possible implementations.

```
int storeFile(InputStream s, String name, Properties metadata);
InputStream openFile(int id);

Set<Integer> search(Properties metadata);
Set<Integer> searchByName(String name);
int search(String name);

Set<Integer> getCollectionItems(int collection_id);
void addToCollection(int collection_id, int item_id);
void removeFromCollection(int collection_id, int item_id);
int createCollection(String name, Properties metadata);

List<FileInfo> list();

String getMetadataProperty(int id, String key);
public Properties getMetadata(int id);
boolean delete (int id);
void clear();
void setMetadata(int id, Properties p);
void setMetadataProperty(int id, String key, String value);
String getName(int id);
```

Database

Currently, a postgresql database with the following structure underlies the repository.

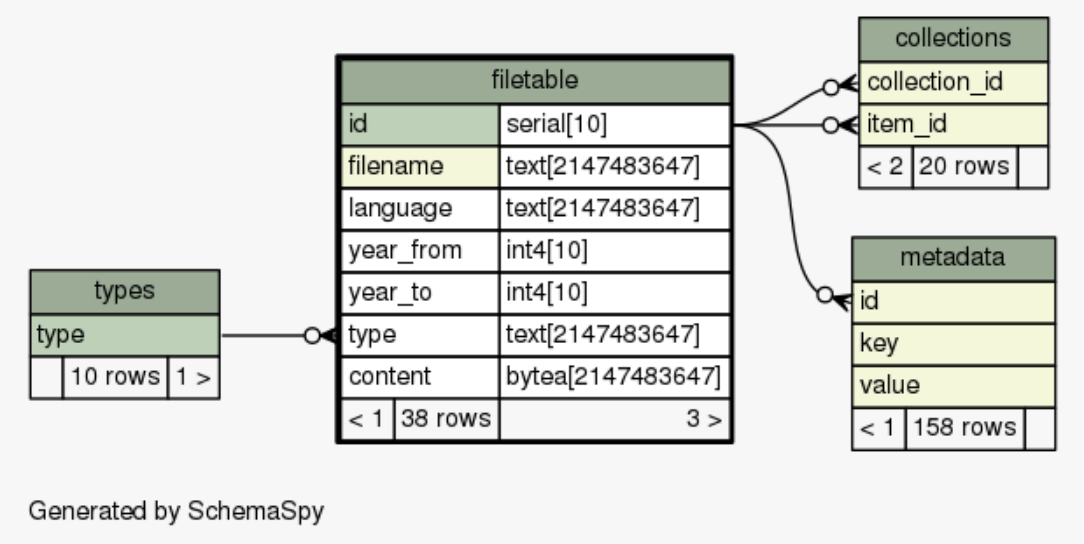

Figure 2: Database structure

Note:

- Storage of file content using the `bytea` data type implies a maximum size of 1G. [8].

[8] cf. `https://wiki.postgresql.org/wiki/BinaryFilesInDB`

- Core metadata attributes are integrated in the main file table, whereas arbitrary other attributes can be stored in the metadata table. The `filename` attribute is constrained to have unique values; the `type` attribute should take values from the `types` table.

REST interface

Action to be taken is controlled by the value of the request parameter `action`.

Repository actions

LIST Lists all resources. Parameters: none. Example request:

```
http://localhost:8080/LMServer/LMServer?action=LIST
```

Response (json array containing, for each entry, id, filename, content length):

```
[
...
{"id":3,"fileName":"resources/CharacterSets/hattem.charset.txt","contentLength":145},
{"id":4,"fileName":"/home/jesse/TUTORIAL-HTR/EXP-HATTEM/TRAIN/LM/TEST-SET/normalizedText.txt"
    ,"contentLength":7341}
..
]
```

GETMETADATA Retrieve metadata for item, by database id. request:

```
http://localhost:8080/LMServer/LMServer?action=GETMETADATA&id=37
```

Response (JSON object containing metadata):


```
{
  "year_to":"1500",
  "witnessYear_from":"1400",
  "medium":"book",
  "pubYear_to":"1500",
  "type":"tei_xml",
  "authorLevel1":"Onbekend",
  "corpusProvenance":"cdrom-mnl",
  "content-length":"9062",
  "createdAt":"Fri Jan 15 13:22:00 CET 2016",
  "witnessYear_to":"1500",
  "titleLevel1":"Collectief lunarium (proza)",
  "pubYear_from":"1400",
  "createdBy":"STORE_FILE",
  "language":"dutch",
  "year_from":"1400",
  "filename":"Dutch/cd-rom-middle-dutch/proza/collectief_lunarium.xml",
  "ipr":"online",
  "domain":"liberal-arts",
  "title":"Collectief lunarium (proza)"
}
```


SEARCHBYNAME Maps resource name to id

```
http://localhost:8080/LMServer/LMServer?action=SEARCHBYNAME&filename=Dutch/cd-rom-middle-dutch/proza/colle
```

```
[37]
```

SEARCH Arguments: JSON object containing search criteria.

```
http://localhost:8080/LMServer/LMServer?action=SEARCH&metadata={language:dutch,year_from:1500}
```

```
[45,31]
```

SETMETADATA Arguments: id (integer), metadata (JSON object)

CLEAR No arguments. Clears the repository.

DELETE Argument: id. Deletes the resource with the given id. Example:

```
http://localhost:8080/LMServer/LMServer?action=DELETE&id=1
```

EXTRACT Argument: id. Download the resource with the given id.

```
http://localhost:8080/LMServer/LMServer?action=EXTRACT&id=37
```

STORE Arguments: a file upload (multipart/form data) and a JSON object containing resource metadata.

```
curl -silent -i -F action=STORE
  -F metadata='{language:english,type:corpus_plaintext,filename:"Bentham/example_bentham_text"}'
  -F name=test -F filedata=@TestData/bentham.train.txt http://localhost:8080/LMServer/LMServer
```

Response: the repository id of the stored file

```
[55]
```

INVOKE Invokes an operation on resources in the repository, producing a new resource. Parameters: `command` specifies the name of the operation. Arguments for the operation are either in separate request parameters or as a JSON object in the request parameter 'params'.

Example: convert and arpa LM to PFSG (probabilistic finite state grammar) representation

```
http://localhost:8080/LMServer/LMServer?action=INVOKE&command=LM2PFSG
   &lm=/home/jesse/TUTORIAL-HTR/EXP-HATTEM/TRAIN/LM/TC/languageModel.lm
   &pfsg=test_hattem_pfsg
```

result is a json object, an associative array mapping (output) parameter names to repository id's.

```
{"pfsg":56}
```

Language model building functions

command: BUILDLM

Construct LM and HTR dictionary from corpus text, parametrized by

Name	type	description
script	name of file in repository	bash script performing the action
conf	name of file in repository	settings file. Contains additional settings: class of tokenizer[9], frequency cutoff.
CHARSET	name of file in repository	character set
CORPUS	name of file in repository	input corpus
OUTPUT	repository pathname for output directory	Output files are produced in a temp. directory, but stored with this prefix in the repository.
languageModel	name	repository filename for produced ARPA lm
dictionary	name	repository filename for produced HTR dictionary
latticeFile	name	repository filename for produced HTK lattice file

[9]Tokenizer implementations include default tokenization and the optimized procedure developed by Alejandro Toselli for the Bentham collection.

```
wget -O- 'http://localhost:8080/LMServer/LMServer?action=INVOKE&command=BUILDLM
       &script=WebContent/LMServerScripts/basicModelBuilding.sh
       &conf=TestScripts/test.settings.sh
       &languageModel=languageModel.lm
       &dictionary=dictionary.txt
       &latticeFile=latticeFile.txt
       &OUTPUT=exampleLanguageModel
       &CHARSET=resources/CharacterSets/bentham_charset
       &CORPUS=TestData/bentham.train.txt'
```

Language model interpolation to combine data from different corpora (for instance in-domain and out-of-domain)

Parameters (for interpolation of two models)

Name	type	description
script	name of file in repository	bash script performing the action
conf	name of file in repository	settings file. Contains additional settings: class of tokenizer[10], frequency cutoff.
CHARSET	name of file in repository	character set
VALIDATION_FILE	name of file in repository	corpus used to calculate optimal interpolation coefficients. Usually the validation subset of an evaluation dataset
MODEL_DESTINATION_DIR	repository pathname for output directory	Output files are produced in a temp. directory, but stored with this prefix in the repository.
languageModel	name	repository filename for produced ARPA lm
dictionary	name	repository filename for produced HTR dictionary
latticeFile	name	repository filename for produced HTK lattice file
COMPONENT_0	name of collection	repository name of collection: language model output folder, containing, besides a component language Model, also the(cleaned) corpus text underlying the model, which is used in the process of creating the combined HTR dictionary
COMPONENT_1	name of collection	similar to above

```
wget -O- 'http://localhost:8080/LMServer/LMServer?action=INVOKE&command=INTERPOLATE_TWO
          &params={script:"WebContent/LMServerScripts/MultipleInterpolationFromConf.sh",
          conf:"WebContent/LMServerScripts/Settings/hattem.interpolate.settings.sh",
          languageModel:"languageModel.lm",
          dictionary:"dictionary.txt",
          latticeFile:"latticeFile.txt",
          MODEL_DESTINATION_DIR:"veryUsefulInterpolatedLanguageModel",
          COMPONENT_0:"/home/jesse/TUTORIAL-HTR/EXP-HATTEM/TRAIN/LM/HATTEM-LM",
          COMPONENT_1:"/home/jesse/TUTORIAL-HTR/EXP-HATTEM/TRAIN/LM/TC",
          CHARSET:"resources/CharacterSets/hattem.charset.txt",
          VALIDATION_FILE:"/home/jesse/TUTORIAL-HTR/EXP-HATTEM/TRAIN/LM/TEST-SET/normalizedText.txt"}'
```

Language model deployment functions

Suggestions from language model

Action: **SUGGESTION**.

Parameters

Name	type	description
lm	name of file in repository	language model file
left	string	left context
right	string	right context
pattern	regular expression	regular expression pattern for focus word, usually prefix
number	integer	maximum number of suggestions to be returned

Example request:

```
http://localhost:8080/LMServer/LMServer?action=SUGGESTION&lm=Demo/cdrom_mnl.lm&left=%20ghi&pattern=.*
```

Response


```
{
    "name":"suggestions",
    "nMatches":101199,
    "entries":[
        {
            "word":"</s>",
            "count":0.11657811050961336
        },
        {
            "word":"sult",
            "count":0.04859437741563634
        },
        {
            "word":"mi",
            "count":0.0419690343700437
        },
        {
            "word":"hebt",
            "count":0.03966075846535435
        },
        {
            "word":"sijt",
            "count":0.036131996950998814
        },
        {
            "word":"en",
            "count":0.03548542575058689
        },
        {
            "word":"selt",
            "count":0.02115640910828042
        },
        {
            "word":"niet",
            "count":0.01870975168847274
        },
        {
            "word":"heren",
            "count":0.015912593926132312
        },
        {
            "word":"die",
            "count":0.013795296830465325
        },
        {
            "word":"u",
            "count":0.013707932288578608
        },
```

```
        {
            "word":",",
            "count":0.012342838532599699
        }
    ]
}
```

Configuration and installation

Prerequisites

1. Linux 64 bits operating system (tested: debian, mint)
2. HTK, SRILM The HTK executables are included in the subdirectory `Tools/HTK/bin.linux` of the war, the SRILM tool binaries and scripts are in `Tools/SRILM/bin` and `Tools/SRILM/bin/i686-m64/`.
3. java, at least version 7
4. Tomcat 7
5. Postgresql database (version 9.1 or up).

Installation

copy the file `LMServer.war` to `/var/lib/tomcat7/webapps` or wherever your tomcat web application directory is located. The source code is in the github repository `https://github.com/JessedeDoes/LMServer`.

Configuration

Database connection properties are given web.xml

```
<init-param>
  <param-name>repositoryConnection</param-name>
  <param-value>{dbHost:svowdb02,dbPort:5432,dbSchemaName:lmserver,dbPasswd:inl,dbUser:postgres}</param-value>
</init-param>
```

Authentication

As the LR server is intended to be more tightly integrated with the Transkribus repository in the near future, a very simple authenticaton method, using HTTP basic authentication, has been implemented. Clients should send basic authentictaion information in the "Authorization" HTTP header, as exemplified by the following command line example.

```
curl -v  --user user:password http://localhost:8080/LMServer/LMServer?action=LIST
* Hostname was NOT found in DNS cache
*   Trying 127.0.0.1...
* Connected to localhost (127.0.0.1) port 8080 (#0)
* Server auth using Basic with user 'user'
> GET /LMServer/LMServer?action=LIST HTTP/1.1
> Authorization: Basic dXNlcjpwYXNzd29yZA==
> User-Agent: curl/7.35.0
> Host: localhost:8080
> Accept: */*
```

User information is stored in the repository database.

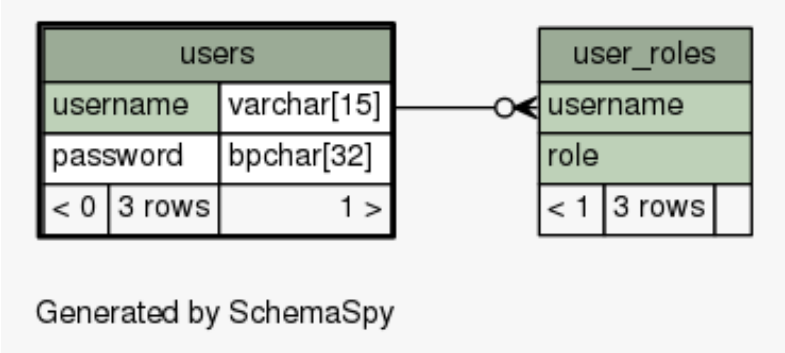

Figure 3: User information in repositoy database

Appendix B: description of resources

Dutch

Resource	distribution	type	language	description
Dutch/cd-rom-middle-dutch	public	corpus (TEI)	Middle Dutch	Corpus of 338 middle dutch texts with metadata. 26 of these belong to the liberal arts domain: Prose: boeck_van_surgien_hs_dh.xml chiromantie.xml collectief_lunarium.xml cyrurgie_hs_br.xml euangelien_vanden_spinrocke.xml fysionomie.xml herbarijs.xml jonghe_lanfranc.xml leringhe_van_orinen.xml livre_des_mestiers.xml medicina.xml liber_magistri_avicenne.xml van_de_vier_elementen_en_complexien.xml Rhyme: anatomie_van_de_mens.xml fysionomie.xml gedicht_over_de_hemeltekenen.xml natuurkunde_van_het_geheelal.xml collectief_lunarium_i.xml collectief_lunarium_ii.xml collectief_lunarium_iii.xml cyromanchie_van_den_pape_van_den_hamme_chiromantie.xml maanzodiologium.xml van_smeinscen_lede.xml heymelijchede_der_heymelijcheit.xml lapidarijs.xml der_vrouwen_heimelijcheit.xml
Dutch/DBNL-artes	restricted	corpus	Middle Dutch	Middle Dutch artes corpus from DBNL, 21 texts
Dutch/HATTEM-rest	public	corpus	Middle Dutch	Part of Hattem manuscript used form language model training
Dutch/Resolutions	public	corpus	17th century Dutch	Materials for language modeling relevant to the Resolutions collection
Dutch/LM	public	language models	Dutch	Extended language models for Dutch collections
Dutch/LM/ExtendedHattemModel	public	language models	Dutch	Extended language models for the Hattem collection
Dutch/LM/ExtendedLeidenModel	public	language models	Dutch	Extended language models for the Leiden collection
Dutch/LM/ExtendedMeermannoModel	public	language models	Dutch	Extended language models for the Meermanno collection
Dutch/LM/ExtendedResolutionsModel	public	language models	Dutch	Extended language models for the Resolutions collection

English

Resource	distribution	type	language	description
English/Bentham	public	corpus	English	Bentham transcripion data used for language model training.
English/ECCO	public	corpus	English	Plain text version of the ECCO corpus used for language model training.
English/ECCO/plainText	public	corpus	English	Plain text version of the ECCO corpus used for language model training. (plain text corpus files)
English/ECCO/selectedSample.lst	public	file list	English	Set of ECCO documents identified as most relevant to the Bentham collection by sample selection methods
English/ECCO/selectedSample.corpus.txt	public	corpus	English	text corpus corresponding to the previous list
English/OLL/	public	corpus	English	Online Library of Liberty edition of Bentham's printed works
English/LM	public	language models	English	Extended language models for English collections
English/LM/bentham_new_tokenization/	public	language models	English	Extended language models for Bentham collection using optimized tokenization
English/LM/ModelsForContestSet/Set0	public	language models	English	Extended language models for Bentham collection: bigram model obtained with sample selection method
English/LM/ModelsForContestSet/Set0	public	language models	English	Extended language models for Bentham collection: trigram model obtained with sample selection method

German

Resource	distribution	type	language	description
German/DTA	restricted	corpus	German	Deutches Text Archiv corpus.
German/Gutenberg	public	corpus	German	Plain text version of German Gutenberg corpus.
German/Gutenberg/Sample	public	corpus	German	Part of previous selected for extended model for the Reichsgericht collection.
German/Reichsgericht	restricted	corpus	German	OCR of book containing relevant Reichsgericht transciptions, used for language modeling.
German/WordLists	restricted	word lists	German	Various word lists from corpora and dictionary headword lists
German/LM/ExtendedReichsgerichtModel/	public	language models	German	Extended language model for Reichsgericht collection using optimized tokenization

Spanish

Resource	distribution	type	language	description
`Spanish/Gutenberg/TXT`	public	corpus	Spanish	Spanish Gutenberg corpus.
`Spanish/Gutenberg/Sample`	public	corpus	Spanish	Part of previous selected for extended model for the Reichsgericht collection.
`Spanish/BVC`	restricted	corpus	Spanish	Corpus of historical Spanish from the Biblioteca Virtual Miguel de Cervantes.[11]
`Spanish/IMPACT-es`	restricted	corpus and lexicon	Spanish	Historical Spanish lexicon and corpus as developed in the IMPACT project