D4.1: Tools for development and deployment of linguistic resources for HTR

Jafar Tanha, Veronica Romero, Jesse de Does

Distribution: Public

TRANSCRIPTORIUM
ICT Project 600707 Deliverable 3.1.2

December31, 2013

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, 2013
Actual date of delivery	December 31, 2013
Date of last update	December 31, 2013
Deliverable number	4.1
Deliverable title	Tools for development and deployment of linguistic resources for HTR
Type	Report
Status &versión	Final
Number of pages	33
Contributing WP(s)	4
WP / Task responsable	INL
Other contributors	UPVLC
Internal reviewer	Joan Andreu Sánchez
Author(s)	Jafar Tanha, Veronica Romero, Jesse de Does
EC project officer	Jose Maria del Aguila
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/

© 2013, The Individual Authors
No part of this document may be reproduced or transmitted in any form, or by any means, electronic or mechanical, including photocopy, recording, or any information storage and retrieval system, without permission from the copyright owner.

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 (D4.1.2) describes the first version of the toolbox for language modeling (deliverable D4.1.1), tools for workflow and deployment of linguistic resources for HTR.

Two main issues of language modeling for HTR are addressed in the project: optimal combination of in-domain and out- domain resources for the construction of N-gram language models, and the construction of HTR dictionaries which relate actually encountered word forms with normalized spellings which are used in N-gram modeling.

In this first release, we have concentrated on aspects of optimalN-gram modeling for the HTR system. One line of work has been the development of tools and workflows for integration of in-domain and out-domain resources. We have considered this problem as a domain adaptation problem, for which we propose a novel approach.In order to evaluate the proposed methods we setup a set of experiments on the Bentham collection and report the results. The results of our experiments show that the proposed methods for sample selection significantly improve the performance of the HTR system.

Another direction is the development of category-based models which exploit document collections with a very regular structure, i.e. a set of Spanish marriage register books covering the period between 1451 and 1905. The results of this approach are presented in this report.

Table of contents

1 Introduction: Language modeling for Handwritten Text Recognition

The TRANSCRIPTORIUM project aims to develop innovative, efficient and cost-effective solutions forthe indexing, search and full transcription of historical handwritten document images, using modern, holistic Handwritten Text Recognition (HTR) technology.Language modeling is an integral part of this type of technology; in order to arrive at a language modeling component which can effectively handle the challenges posed by historical language, special resources and approaches are necessary. Accordingly,in work package 4, we develop linguistic resources to support the HTR system, andstudy the effect of their application.
The current deliverable describes methods and tools which have been developed in the first year of the project. We specifically concentrate on the domain adaptation problem for the HTR system, where the main challenge is how out of domain data can be used in HTR system (cf. for instance [Guillem Gascó, et. Al., 2012]), and on a category-based approach to language modeling for documents with a regular semantic structure.

1.1 Challenges of language modeling for HTR in tS

Historical documents pose problems for language modeling in two ways:

1. It is usually difficult to get hold of more than a very limited amount of corpus material that is relevant for the type of language that needs to be modeled.
2. (Historical) spelling variation exacerbates this problem: with a limited training corpus, even more words and word combinations will be unknown.

To address these problems, TRANSCRIPTORIUM follows two lines of research:

1. Use a combination of in-domain and and out-of-domain language data to estimate language models.
2. Develop and deploy variation lexica to overcome spelling variation problems.

1.2 Lexical and N-gram modeling

Language modeling in TRANSCRIPTORIUMconsists of *lexical modeling* and *sequence or N-gram modeling.* To reduce the data sparseness problem, N-gram models are trained on text corpora in normalized form. The process of obtaining normalized word forms is called *text normalization.* The *lexical model* defines the probability that a normalized word is “realized” in actual text by a certain “witnessed” form.
N-gram models are used to assess the probability of a (normalized) word occurring in a certain context. In TRANSCRIPTORIUM, we use N-gram models of *normalized* word forms. These two componentsdefine a language model for unnormalized word forms by

$$p(w|h) = p_{lexicon}(w|W) * p_{LanguageModel}(W_i|W_{i-1} \ldots W_{i-n+1}),$$

where *w* denotes an unnormalized word, *h* (history) the words preceding it in the text, and *W* denotes a normalized word form.This assumes that the normalized form is uniquely defined for each witnessed word *w.* If the normalization process is such that two or more normalized word forms may correspond to one witnessed word form, as may occur with the more complex normalizations we use for historical text, this becomes

$$p(w) = \prod_{i=1}^{l} \sum_{W_i \in Normalizations(w_i)} p(w_i \mid W_i) \cdot p(W_i \mid W_{i-1} W_{i-2} \ldots W_{i-n+1}) .$$

1.3 Lexical modeling

Lexical knowledge is integrated into the TRANSCRIPTORIUM HTR system in the form of a list of finite-state word models. The model of each word represents, in terms of character-level tokens (characters, digits and symbols), all the expected forms the considered word may have been written. In the simplest case, a word model is just a short list of alternative word-forms, each one accounting for the optional capitalization of some of its characters. Lexicon models can be automatically learned from sufficiently large amounts of training (transcribed) text from the considered documents or from other similar sources. However, huge amounts of text would be needed to learn all the word forms of the unrestricted vocabulary underlying a typical document or document collection. This situation can be alleviated if a vocabulary can be compiled using a prior (human expert) knowledge about the kind of text expected to appear in the documents of interest. In TRANSCRIPTORIUM, a specific effort is devoted in WP4 to search, compile and adequately transform lexical resources for their use in the HTR decoding of the documents to be considered in the project.
In order to avoid data sparsity in N-gram modeling (cf. the next section), the word forms found in actual text need to be reduced to standard forms, of which we then calculate the N-gram probabilities to obtain language models. As previously discussed, vocabularies for HTR are not flat word lists which can be easily extracted from plain text corpora. We need to relate word forms with the normalized spellings which are used in N-gram modeling. This reduction to normal form is also necessary to relate in-domain and out-of-domain data, which may use different spellings, to each other. Part of this data can be obtained automatically (like uppercase-lowercase mapping, or completely productive spelling variations), but supervision is necessary for less obvious cases - for instance, classifying variant forms for medieval language is far from obvious (cf . e.g. [Kestemont 2010]). E.g. variant forms of Dutch *uiterlijk* like *uytterlijc, uyterlijk, uiterlyk, uyterlijck, uiterlijk, uiterlick* are predictable with some knowledge of historical spelling on the basis of a few frequent spelling patterns, but other cases are not so simple.

1.3.1 Dealing with (historical) variants

In medieval and early modern text, spelling variants can be especially prolific. Some possible realizations of the Dutch word “pelgrim” (pilgrim) are listed below:

> pelegrime, pelgrime, peelgrime, peilgrime, pilgrime, pellegrime, peregrime, pelegrom, pelgrom, peelgrom, peilgrom, pilgrom, pellegrom, peregrom, pelegrum, pelgrum, peelgrum, peilgrum, pilgrum, pellegrum, peregrum, pelegrim, pelgrim, peelgrim, peilgrim, pilgrim, pellegrim, peregrim, pelegrijn, pelgrijn, peelgrijn, peilgrijn, pilgrijn, pellegrijn, peregrijn, pelegrijm, pelgrijm, peelgrijm, peilgrijm, pilgrijm, pellegrijm, peregrijm, pilgerijm, pelgerijn, pilgerijn, pillegram, pilgram, pylgram, pylgrym

Our approach to this problem is to:

1. Use language models on the level standardized word forms, or of a combination of standardized lemma form + part of speech.
2. Develop variation lexica, exploiting lexical resources like scholarly dictionaries and productive spelling variation patterns. Such a lexicon would contain, for instance, all alternative spellings for pelgrim.

Note that in this approach, the normalized word form is no longer necessarily uniquely determined by

the witnessed word form as found in actual text, and nontrivial natural language processing may be needed to perform text normalization.
This approach requires a workflow for resource development using annotated corpora, roughly as follows:

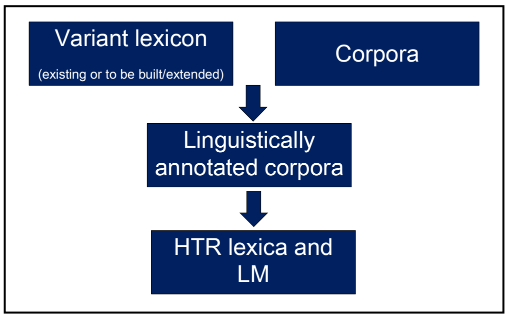

Variant lexicon
(existing or to be built/extended)
Corpora
Linguistically annotated corpora
HTR lexica and LM

Figure 1-1: lexica from linguistically annotated corpora

In the Bentham data we have been dealing with so far, the variation problem is not so much in evidence. We have not yet been able to experiment with HTR for the older Dutch and German data, where spelling variation of some kind affects almost all words found in the text.

A relevant line of research are factored language models and other approaches to take inflectional morphology into account [Bilmes & Kirchoff 2003, Chanuneau et.al. 2013].Note that the category-based models of section 5 can be considered an extreme form of text normalization.

1.3.2 Dealing with hyphenated words in the lexicon

Typically, the TRANSCRIPTORIUM historical data containsmany hyphenations, which break words into twoparts, as well as instances of words which are split without any explicit graphical indication:

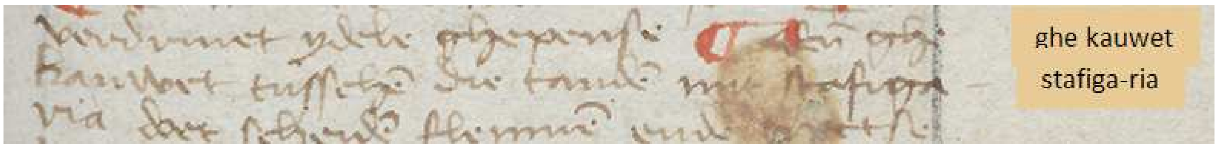


Figure 1-2: two consecutive lines ending with incomplete words. The second case (stafiga-ria) has an explicit hyphen, the first case (ghe|kauwet) is not marked in any way. (Hattem manuscript)

Currently, the language modeling in the TRANSCRIPTORIUM HTR system does not in any special way deal with hyphenated words. If the parts have of a hyphenated word have not been encountered in the training set for the language model, they lead to unknown words and N-grams. Furthermore, there is no sharing of probability information between hyphenated and non-hyphenated occurrences of words.
We propose that hyphenation information can be best accounted for by the lexical model, leaving the N-gram model untouched. Both for hyphenated and for non-hyphenated line-final word forms, we consider the line break part of the witnessed word[1] and list (at least virtually), occurrences of words at the end of a line and hyphenated words as variant realizations in the lexicon, cf. the following example:

```
'HYPHEN'→
{
('hyphen'→ 0.8), ('Hyphen' → 0.1),
('hyphen|' → 0.075), ('hy-|phen' → 0.02),
('hy|phen' → 0.005)
```

[1]This may seem a bit artificial, but modeling the line break as a separate token is even less attractive.

}

Of course, the frequencies listed can never be obtained by mere counting, and the probabilities involved will have to be parameterized by a few global estimates of a few obviously highly text-dependent, and possibly word length or word-dependent, probabilities.

1) The relative probability p_f of finding a word at the end of a line (something like 1/ (average line length)).
2) The probability p_h of encountering a given word in some hyphenated form, using a hyphenation sign.
3) The probability p_s of encountering a word split across lines, without explicit hyphenation.
4) For each hyphen-able word, the probabilities of its different hyphenations.

1.3.2.1 How to integrate support for hyphenation in the HTR system?

There are several studies on dealingwith this problem, cf. for instance [Lavrenko et.al. 2004], who suggest to implement a hyphenation detector in the preprocessing stage and to concatenate word images when a hyphen is detected. To overcome this problemin our setting, one possible solutioncould be:

- The feature files from which the actual recognition takes place after feature extraction will have to be concatenated[2] for a text region or paragraph.
- The line break is considered a special character; there needs to be a character HMM which recognizes this character in multiline feature files containing.
- The hyphenation lexicon will have to constructed by full expansion.

1.4 N-gram modeling

In order to model sequences of words, statistical N-gram models are used in many natural language processing applications such as speech recognition, machine translation or handwriting text recognition. The role of a language model is to score and/or prune out unlikely word sequence hypotheses which would otherwise be compatible with the underlying HTR lexical and morphological (HMM) models.

Language models are generally obtained from plain text corpora, which ideally are linguistically close to the language of the document collection which is being processed. Large corpora are needed in order to avoid overfitting. In language modeling for historicaldocuments this is a real risk. since usually only insufficient amounts of data may be available for a specific document or collection [Wuthrich et.al.2009].

Current approaches to HTR often use just a single-resource approach, where the models are obtained from material taken from a part of the target document or collection. InTRANSCRIPTORIUM, we build language models from a combination of language resources, consisting of larger "*out-of-domain*" data sets and smaller "in-domain" corpora. *Out-of-domain* data sets can more easily be obtained, since there currently are many free available plain text corpora that can be used for this purpose[3]. *In-domain* data can be obtained as a document image collection is being transcribed. The in-domain data can be

[2] Even without the hyphenated word issue, this is probably better because normal language modeling also breaks down at line breaks in the current setup.

[3] For historical material, even out-of-domain corpora may be hard to come by.

used for tuning the out-of-domain language model. One possible approach is *data selection* (try to exploit the best possible subset of the out-of-domain data), for instance by usingentropy-based measures on the sentence level [Moore & Lewis 2010]. Many other solutions for data selection and language model adaptation have been suggested in the speech recognition and machine translation communities.

Some document collections, like baptismal or marriage register books, exhibit a very regular structure and are characterised by the large amount of proper names. These peculiarities can be exploited in order to improve the recognition process. Language models that strictly account for the syntactic structure of the lines in the studied documents can be defined, notably restricting the search space [Romero et. al. 2011]. TRANSCRIPTORIUM is investigating the production and use of language models for document collections of this type. Most specifically, the use of category-based language models has been studied [Romero & Sanchez 2013].

1.5 Evaluation of language models

To select the best possible language models, or to judge the degree to which a document (or some other sequence of words) 'fits' a language model, we need evaluation criteria for language models.
The most obvious evaluation criterion of a language model *M* used in conjunction with a HTR system is simply the word error rate (WER) or character error rate (CER) observed when the HTR uses *M*on some evaluation data set D. Such an evaluation, however, is costly, and only possible in a situations where have both images and ground truth transcription for D. It cannot readily be used for informative data selection methods like the ones introduced in section 4. Hence we also need purely text-based methods.
There is an extensive literature on the evaluation of language models(e.g. [Chen et. al. 1998]). The degree to which a word sequence $W=(w_1 \dots w_n)$'fits' a language model with probability function *p*is usually measured by *cross-entropy* or *perplexity*, defined by

$$CE(W,p) = \sum_{i=1}^{N} \frac{1}{N} \log_2 p(w_i)$$

In many situations, a positive relation between recognition rate and perplexity can be observed. For instance [Klakow and Peters2002] find a linear relation between log(PPL) and log(WER) for a fixed speech recognizer. In other cases, the relation is not so obvious.
When comparing language models with different text normalization strategies, it should be taken into account that the real evaluation should take place on the level of unnormalized words.Currently, we have been workingwith the Bentham Data, where the text normalization strategy is not of prime importance, and we can afford to base our evaluations on the output of standard language modeling toolkits (like SRILM, [Stolcke 2011]) run on normalized text. A different approachwill be necessary when we will need to compare different normalization strategies on the older TRANSCRIPTORIUM data.

1.5.1 Taking Out-Of-Vocabulary (OOV) words into account in model evaluation

Unfortunately, the above definition of perplexity is insufficiently explicit in the case the text contains words which are not accounted for by the language model. An often-used workaround is to replace all unknown words by an <*UNK*>symbol, for which a fixed probability $p_{unk}$is chosen, for instance estimated from the frequency on infrequent words in the training corpus. An undesirable side effect of this is that a model with a larger vocabulary of lower-frequency words, which in practice may perform better when deployed in the HTR system, can end up having worse perplexity than one with a smaller vocabulary, estimated from a smaller training corpus.In practice, this means that perplexity is best used to compare models with the same vocabulary, which is an extremely undesirable restriction.For several approaches to the OOV problem, originating from different domains, cf. for instance [Scharenborg et.al. 2007].

Let $|V|$ be the number of words (#tokens) of in the evaluation data and $|OOV|$ be the number (in #tokens) of out-of –vocabulary words. Now, there are several options:

- Assign a fixed low probability to each unknown word (SRILM).
- Allow the user to set this probability (IRSTLM)[4].
- Use *adapted perplexity*, (*APP*, [Ueberla 1994]): if o is the number of *different* unknown words occurring in the test set, $ALTP = CE - o * \log_2(|\mathrm{OOV}|)$, $APP = (2^{ALTP})^{(-1/|V|)}$.
- Using ranking functions combining |OOV| and perplexity, as we will proceedin the sample selection methods of section 4. We define three distinct criteria:
    1. (additive criterion)(log PPL) + $|OOV|/|V|$.
    2. (multiplicative criterion) (log PPL) * $|OOV|/|V|$.
    3. (average word log probability, with *p(unknown)* set to 0) (1/log PPL)*(1-$|OOV|/|V|$)[5].

The multiplicative criterion seems to do best in practice, cf. the HTR results in section 7.1.

2 BasicTRANSCRIPTORIUM workflow for language modeling

The current TRANSCRIPTORIUMholistic handwriting recognizer is a HMM-based recognizer implemented on top of the well-known HTK hidden Markov model toolkit.[6]

To use language models with the recognizer, one needs two resources:

1. A (bigram[7]) language model, in HTK lattice format. This models sequences of *normalized* tokens.
2. A dictionary file, specifying the relation between normalized words and character strings, with relative frequencies for each possible realization.

2.1 Preprocessing and dictionary creationfrom corpus text

In order todevelop a language model and a HTR dictionary for a given HTR system from a training corpus, two preprocessing steps are needed:

1. *Text cleaning* removes characters that are not recognized by the HTR system (words containing these characters are mapped to a special "unknown" token).

[4]http://sourceforge.net/apps/mediawiki/irstlm/index.php?title=User_Manual: Language models have to cope with out-of-vocabulary words, that is internally represented with the word class *_unk_* . In order to compare perplexity of LMs having different vocabulary size it is better to define a conventional dictionary size, ordictionary upper bound size, trough the parameter (-*dub*).

[5]Another possibility the average percentage of words that a purely LM-based guesser will get right. This can be estimated by

$P_{guess}(w|h) = 1/ppl$ for in-vocabulary words

$P_{guess}(w|h) = 0$ for OOVwords

With an OOV rate (# OOV tokens / # tokens) of α, this simple gives (1- α)/ppl.

[6] http://htk.eng.cam.ac.uk/

[7] Trigram models can be used for pruning word graphs, but not directly during recognition.

2. *Text normalization* applies a normalization to each token in the text in order to reduce data sparsity for N-gram modeling.

The result of this phase is twofold:

1. A normalized text corpus.
2. A HTR dictionary relating normalized to actually encountered word forms.

This consists of two steps: the first step to prepare data for HTR system is to clean the resources based on a set of characters. This set includes all characters that are applicable by HTR system, see Figure 1. We have therefore developed several tools to prepare the resources. This section will address the preparation steps and all developed and used tools. The second step is to build a language model (LM) and the related dictionary.

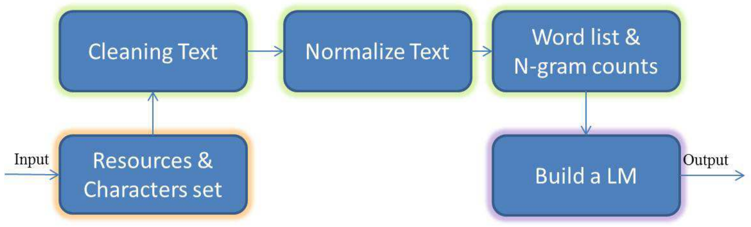


Figure 2-1, Workflow to generate an N-gram LM

2.1.1 The HTR dictionary

The HTR dictionary relates normalized words to actually encountered word forms. It is created during text normalization, according to the text format expected by HTK, see Figure 2.

2.2 *Building language models*

We use the well-known SRILM toolkit[8] to train the n-gram models from the training corpus. The developed scripts are presented in the appendix. Figure 1 shows the general steps to generate a Language Model (LM).

2.2.1 Conversion to HTK lattice file format

Now we can use the generated LM in HTR system. The only remaining step is to convert to the lattice file format required by HTK. This is accomplished by the HTK took `Hbuild`. Figure 2 summarizes the required steps, see Figure 3.

[8]http://www.speech.sri.com/projects/srilm/. SRILM is a toolkit for building and applying statistical language models (LMs), primarily for use in speech recognition, statistical tagging and segmentation, and machine translation.

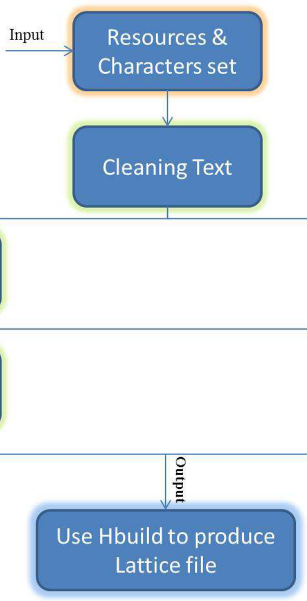

Input
Resources & Characters set
Cleaning Text
Word list & Word frequency
Normalize Text
Make a Dictionary
Build a LM
Output
Use Hbuild to produce Lattice file

Figure 2-2, Workflow for LM and dictionary creation

3 TRANSCRIPTORIUMdata used in this report

3.1 English language data

We mainly use the TRANSCRIPTORIUM English-language corpus data. This consists of:

- the Bentham corpus of transcribed manuscripts (about 15.000 pages and5m words).
- the public part of the ECCO (Eighteenth Century Collections Online)[9], about 70m words.

With these two corpora, we make a two-level in-domain/out-of-domain distinction:

- The ECCO corpus is considered as out-of-domain resources.

Within the set of Bentham transcripts we distinguish the of Batch 1 ground truth transcriptions as an in-domain resource and the rest as out-of-domain.

Table 3-1: English data sets

Resources	Size	File Name
Bentham In "Batch 1"	57.7 (kb)	train_0.txt.norm
Bentham Out	38.3(mb)	NotBatch1.lpl.txt
ECCO	425.8(mb)	ECCO_ Folder
Test set	6.5 (kb)	test_0.txt.norm

3.2 Spanish language data

We have used the publicly available ESPOSALLES database[10][Romero et. al. 2013].

This database has a total of 173 pages that contain 5,447 lines grouped in 1,747 licenses. The complete annotation contains around 60,000 running words from a lexicon of around 3,500 different words.To carry out the experiments we have used the standard partition proposed in [Romero et. al. 2013], consisting of seven consecutive blocks of 25 pages. Table 2 summarizes the statistics.

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

[10] The corpus is publicly available from: http://www.cvc.uab.es/5cofm/groundtruth

Table 3-2, Basic statistics of the different partitions for the database ESPOSALLES

Number of:	*P0*	*P1*	*P2*	*P3*	*P4*	*P5*	*P6*
Pages	25	25	25	25	25	25	23
Licenses	256	246	246	249	243	255	252
Lines	827	779	786	768	771	773	743
Run. words	8,893	8,595	8,802	8,506	8,572	8,799	8,610
OOV	426	374	368	340	329	373	317
Lexicon	1,119	1,096	1,106	1,036	1,046	1,078	1,011
Characters	48,464	46,459	47,902	45,728	46,135	47,529	46,012
Semantic labels	3,155	3,026	3,162	2,988	2,948	3,069	3,038

4 Combining in-domain and out-of-domain resources: TRANSCRIPTORIUM approaches

InTRANSCRIPTORIUM, we investigate how to obtain good language models from a combination of language resources, consisting oflarger out-of-domain data sets and smaller in-domain corpora.We investigate a number of possibilities to combine resources:

- *Interpolation* of models estimated from different resources.
- *Sample selection:* trying to exploit the best possible subsets of the out-of-domain data.
- *Topic modeling:*obtain language models specifically adapted to the mix of topics occurring in a document.

4.1 Interpolationof Language Models

One way of combining resources is interpolation of language models estimated from these resources. This is a well-established technique, which is supported by the SRILM toolkit. We will use this as a baseline for evaluating the more advanced techniques introduced later on.

For two models $LM_1$and$LM_2$with respective transition probabilities p_1 and p_2, the interpolatedmodel LM_λ (depending on an interpolation parameter λin [0, 1]) will compute the probability of a word given its history by

$$p_\lambda (w|h)= \lambda\, p_1(w|h) +(1- \lambda)p_2(w|h)$$

In general, interpolation of n language models requires n-1parameters. To interpolate two or more LMs, we have to estimate the optimal weights for each LM from a development set. Figure 4-1 addresses the required steps for interpolation of two LMs.

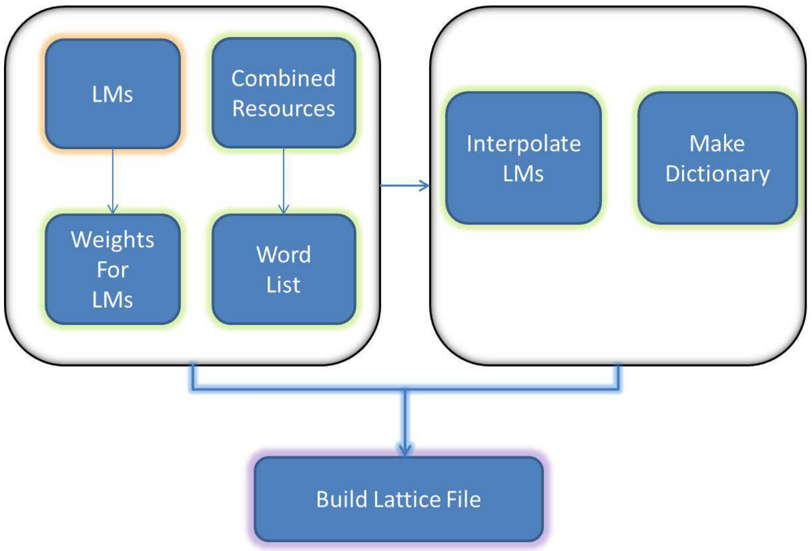


Figure 4-1, Interpolation of LMs

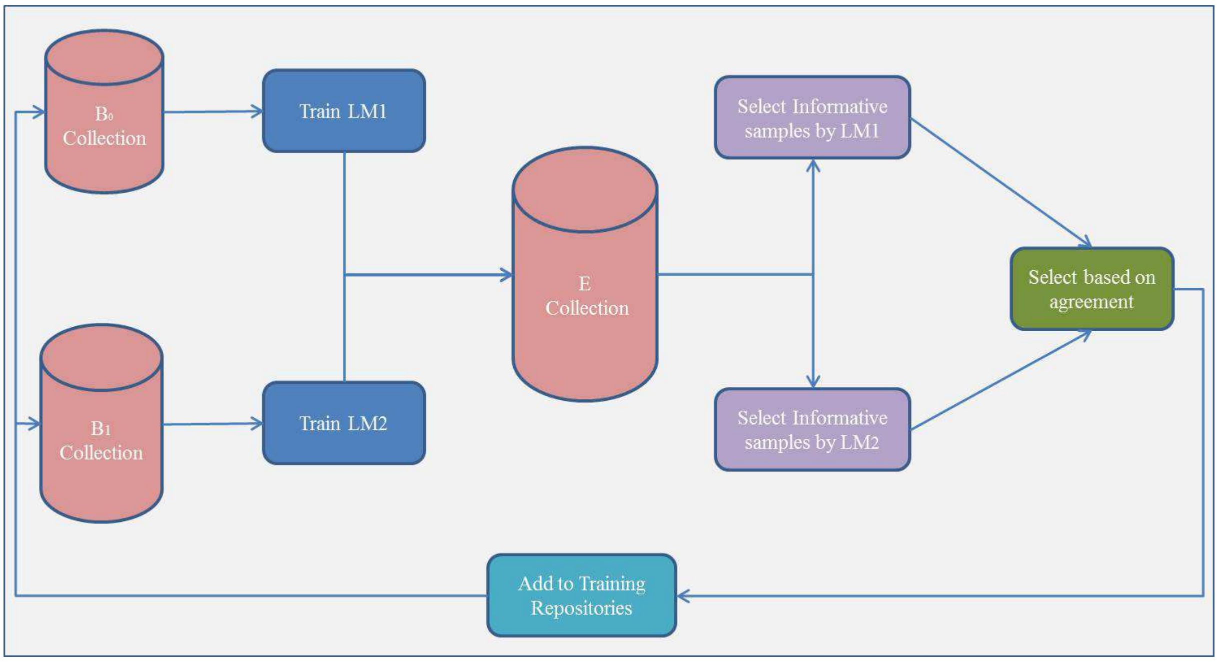
4.2 *Intelligent Sample Selection*

In this section, we study the domain adaptation problem from the point of view of intelligent sample selection, using both in-domain and out-of-domain resources, in order to improve the performance of the HTR system. Studies in this direction show that the performance of recognition and automatic translation systems can be improved by employing language models which have been trained based on larger corpora, see [Gao et. al. 2000], [Moore & Lewis 2010], and [Gascó et. al. 2012]. However these larger corpora have to be related to the in-domain data, which is often difficult to find in practice. Therefore the main goal of this work is to introduce and propose methods to find and select informative data from the out-of-domain resources. There are several studies on the domain adaptation problem for different naturallanguage processing tasks, machine translation, information retrieval, and speech recognition [Moore & Lewis 2010], and [Gascó et. al. 2012].

In [Gao et. al. 2000], a text retrieval approach has been proposed for domain adaptation problem.The main idea of this approach is to avoid items specific to the out-of-domain data by removing n-grams likely to be infrequent in new documents, based on a partitioning of the training data.

Moore and Lewis [Moore & Lewis 2010] have proposed a cross-entropy based approach to sample randomly from the out-of-domain data using perplexity as a main criterion. More recently, Gascó et.al. [Gascó et. al. 2012] address a data selection approach from out-of-domain corpora by approximatingthe probability of an in-domain corpus.

The abovementioned methods haveyielded some performance improvements in the addressed NLP task, but the ranking criteriado not take the relationship between OOV and perplexity into account, which may result in poor sample selection. To overcome this problem, we propose two iterative approaches for sample selection, based on ranking criteriataking both OOV rate and perplexity into account (cf. section 1.5.1). Our proposed methods take an approach which is inspired by co-training and self-training of ensemble semi-supervised learning approaches in machine learning [Tanha 2013]. Co-training is a semi-supervised approach for the exploitation of informative unlabeled data, in which two or more classifiers are trained from iteratively growing training sets, in such a way that instances from the unlabeled data are added to the training data of one of the classifiers when it can be labeled with sufficient confidence by the other classifier(s). Traditionally, co-training depends on complementary*views* of the data, but it has been shown that there are other scenarios by which classifiers can learn from each other [Nigam & Ghani 2000, Tanha et.al. 2011]. The main advantages of this iterative approach is that it gradually exploits informative unlabeled data and assigns labels to them. We fit our problem in this framework and propose two iterative approaches. We consider each language model, which has been trained on a corpus, as a classifier and use it for ranking the out-domain data. Figure 4-2 shows a general overview of the proposed methods.



B0 Collection
Train LM1
Select Informative samples by LM1
E Collection
Select based on agreement
B1 Collection
Train LM2
Select Informative samples by LM2
Add to Training Repositories

Figure 4-2, General overview of the proposed approach

The general setting of the proposed method is that we start from a set of three corpora B_0, B_1 and E, such that $\#|B_0| < \#|B_1| < \#|E|$, and the B_1 material is (much) more similar to the B_0 material than the material from E.B_0 and B_1 may arise from a partition of a corpus B. The proposed task is to select the most informative material from E. Unlike other, sentence-based approaches [Moore & Lewis 2010; Gascó et. al. 2012], we will apply sample selection on the document level.
We propose two iterative algorithms to select an informative subset of high-confidence resources from the *E*collection, based on the *agreement* or *disagreement* between two language models LM_i (i=1,2) trained on (extensions of) B_0 and B_1 respectively.
In our setting, B_0 is a small set of in-domain Bentham data (The "Batch 1" set), and B_1 consist of the remaining part of the Bentham collection. The really out-of domain corpus E=ECCO consists of 2207 documents from different topics.

4.2.1 Agreement-based Approach

We first propose an agreement-based co-training approach (*Agree-Co*) for sample selection from out-domain data. The general idea of *Agree-Co* is to use the agreement between two LMs for selecting a set of high-confidence resources from the out-domain data.
In *Agree-Co* two LMs are first trained on B_0 and B_1. The trained models then use the E (=ECCO) collection as test set; each document from E is evaluated in terms of LM_j(j=0,1) by means of a scoring criterion C_j,which is a function of perplexity and number of OOV's[11]. The scoring criterion usedhas high impact during the sample selection step. The evaluated documents have different number of OOV and therefore using only perplexity to rank cannot be a suitable criterion. We propose three criteria for ranking, as briefly mentioned in Section 1.5.1. For eachLM_j(j=1,2), the algorithm then sorts the set of documents by the value of the scoring criterion, and selects a set of high confidence resources H_j,

[11]We have estimated the correlation between PPL and OOV of the resources. Based on our estimation the correlations are r_{in}= -0.23 and r_{out}= -0.04.

determined as the *p*% best scoring documents with respect to the scoring criterion. The intersection of the two H*j*is then added to the original training sets. After this, the training process is repeated until some stopping condition is reached. The threshold parameter in the algorithm is calculated by averaging the highest scores of the selected subsets. The pseudo-code of the *Agree-Co* algorithm is presented in Algorithm 1.
In the experiment section, we will explain several approaches to use the selected resources for improving the HTR system. We will further address the number of iterations there.

```
Algorithm for domain adaptation Agree-Co($B_0$, $B_1$, $E$, max_iterations, threshold)
Initialize:
t=0 ,  conf=0;
        select a resource scoring criterion $C$
Begin
While (t<max_iterations&&conf<threshold)
Begin
      Build $LM_0$ and $LM_1$ from $B_0$ and $B_1$
For each $R_i$ in $E$do
Begin
         Compute $C_0(R_i)$ and $C_1(R_i)$
End
      $H_0$ := a high-confidence subset of $E$, selected by best values w.r.t.$C_0$
      $H_1$ := a high-confidence subset of $E$, selected by best values w.r.t.$C_1$
Assign a value to confby averaging the worst values of $C_0$and$C_1$on$H_0$resp. $H_1$
$S_t := H_0 \cap H_1$
$B_0 := B_0 \cup S_t$
      $B_1 := B_1 \cup S_t$
$E := E \setminus S_t$
      t := t+1
End // while
   Output:  Selected Resources for domain adaptation and $B_0$ and $B_1$.
End/ / Algorithm
```

Algorithm 4-1, pseudo-code of the Agree-Co Algorithm

4.2.2 Disagreement-based Approach

In the *Agree-Co*Algorithm the *agreement* between two LMs is used to select a set of informative resources to be added to the original training resources.
Another approach is to exploit the *disagreement* between two LMs.The resulting Disagreement-based Co-training (*Disagree-Co*) algorithm differs from *Agree-Co* in the way that the training material for the language models is extended. Whereas*Agree-Co* adds the intersection of the high-confidence sets H_j *(j=0,1)*to the training sets for both language models, what *Disagree-Co*adds to the training material for the first model is the set of resources which are in the high-confidence set for the first model, but *not* in the high-confidence set for the second model, and vice versa.The scoring criteria used in *Agree-Co* and *Disagree-Co* are the same. The pseudo-code of *Disagree-Co*is presented in Algorithm 2.

```
Algorithm for domain adaptation Disagree-Co($B_0$, $B_1$, E, max_iterations, threshold)
Initialize:
t=0 , conf=0;
        select a resource scoring criterion C
Begin
While (t<max_iterations&&conf<threshold)
Begin
    Build $LM_0$ and $LM_1$ from $B_0$ and $B_1$
For each $R_i$ in E do
Begin
        Compute $C_0(R_i)$ and $C_1(R_i)$
End
    $H_0$ := a high-confidence subset of E, selected by best values w.r.t. $C_0$
$H_1$ := a high-confidence subset of E, selected by best values w.r.t. $C_1$
Assign a value to conf by averaging the worst values of $C_0$ and $C_1$ on $H_0$ resp. $H_1$
$S_0 := H_0 \setminus H_1$
$S_1 := H_1 \setminus H_0$
$B_0 := B_0 \cup S_0$
    $B_1 := B_1 \cup S_1$
$E := E \setminus (S_0 \cup S_1)$
    t := t+1
End // while
  Output:   Selected Resources for domain adaptation.
```

Algorithm 4-2, Pseudo-code of the Disagree-Co Algorithm

4.3 Using topic modeling

An obvious approach to adaptive language modeling is to try to develop language models and lexica specific for the subject matter of the text. Since manual classification of topics is cumbersome and highly subjective, this line of research only took off after unsupervised topic modeling, notably LDA, was developed in the early 2000's.

Latent Dirichlet Allocation (LDA, [Blei et. al. 2003][12]) estimates topic-dependent unigram language models from a set of documents. Several approaches have been suggested to go beyond isolated words[13].

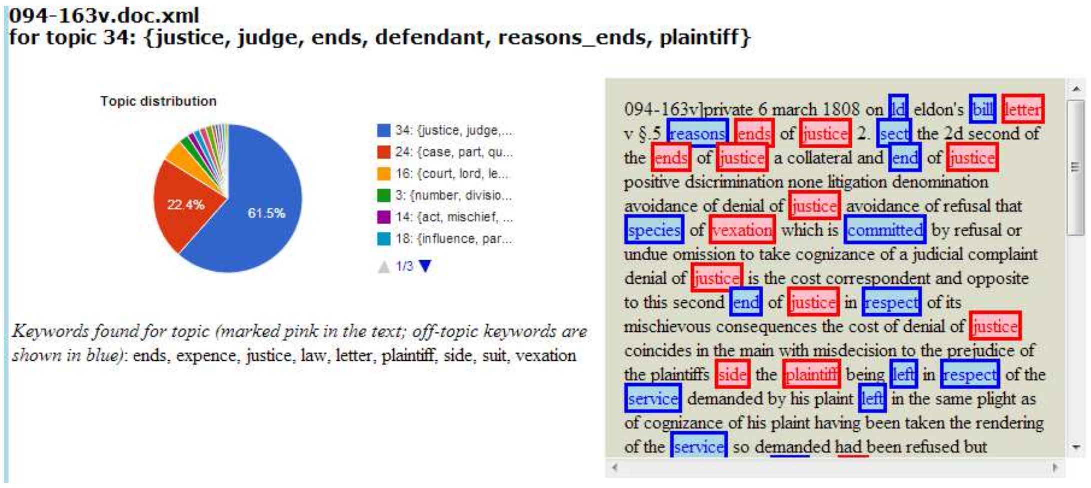
The basic scenario for topic-based language model adaptation is:

1) Compute an n-gram topic model from a corpus of in-domain and out-of-domain data.
2) To obtain an adapted LM for a document D, obtain, either from in-domain training data or from a first processing round, an inferred topic distribution φ for D.
3) The language model is the specialization of the general topic model to the inferred topic distribution φ.

[12] Very useful is D. Mimno's LDA Bibliography An exhaustive list of LDA-related resources (incl. papers and some implementations)

[13]Cf. for instance Wallach 2008, Wang & MacCallum, Lindsey ea 2012.

The basic intuition is that, by applying the right kind of smoothing in the topic modeling procedure, we can obtain models which exploit the relevant parts of the corpus for topic-specific vocabulary and phrases, but still profit from the whole corpus for general, topic-neutral words and phrases – which would not be the case in a sample selection based procedure.
We have experimented on the Bentham and ECCO data with the Topical N-grams model, which has an implementation in the Mallet statistical NLP platform. Although it gives intuitively plausible keywords and multiword key phrases, it does not yet give us the necessary implementation of an "inferencer" (which estimates a topic distribution for and unknown document) and an "estimator" (which computes the perplexity of a document given a topic distribution); moreover, we have doubts about the suitability of the TNG model, which targets keyphrase extraction, for language modeling purposes. Other approaches might be more suitable, but before we invest time in implementing them, we need some confirmation that we are on the right track.

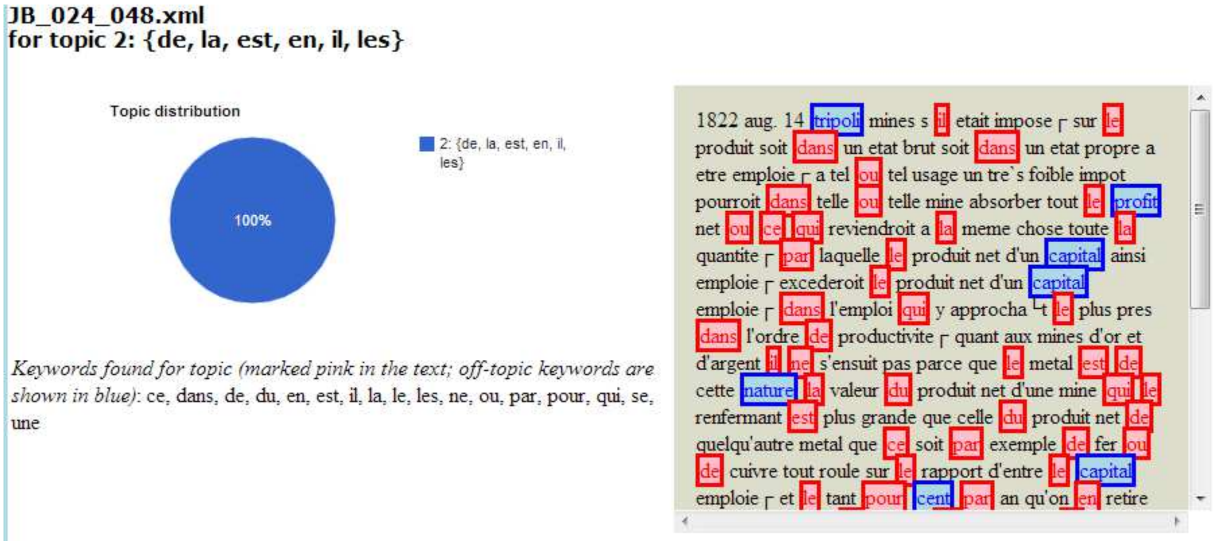


094-163v.doc.xml
for topic 34: {justice, judge, ends, defendant, reasons_ends, plaintiff}

Keywords found for topic (marked pink in the text; off-topic keywords are shown in blue): ends, expence, justice, law, letter, plaintiff, side, suit, vexation

094-163v]private 6 march 1808 on ld eldon's bill letter v §.5 reasons ends of justice 2. sect the 2d second of the ends of justice a collateral and end of justice positive dsicrimination none litigation denomination avoidance of denial of justice avoidance of refusal that species of vexation which is committed by refusal or undue omission to take cognizance of a judicial complaint denial of justice is the cost correspondent and opposite to this second end of justice in respect of its mischievous consequences the cost of denial of justice coincides in the main with misdecision to the prejudice of the plaintiffs side the plaintiff being left in respect of the service demanded by his plaint left in the same plight as of cognizance of his plaint having been taken the rendering of the service so demanded had been refused but

Figure 4-3, topic distribution for a document

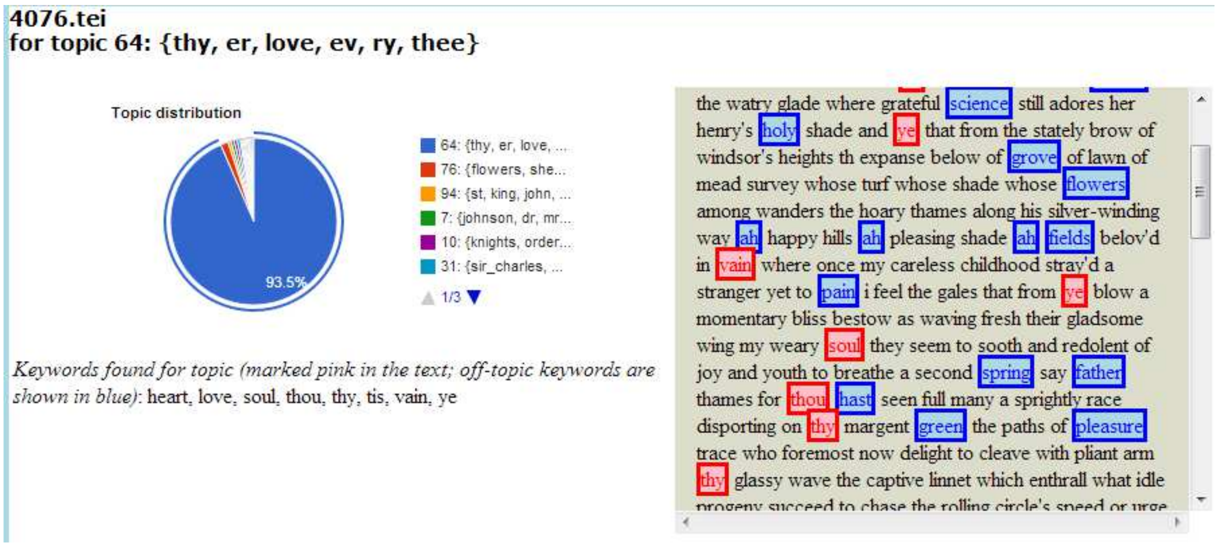


JB_024_048.xml
for topic 2: {de, la, est, en, il, les}

Keywords found for topic (marked pink in the text; off-topic keywords are shown in blue): ce, dans, de, du, en, est, il, la, le, les, ne, ou, par, pour, qui, se, une

1822 aug. 14 tripoli mines s il etait impose ┌ sur le produit soit dans un etat brut soit dans un etat propre a etre emploie ┌ a tel ou tel usage un tre`s foible impot pourroit dans telle ou telle mine absorber tout le profit net ou ce qui reviendroit a la meme chose toute la quantite ┌ par laquelle le produit net d'un capital ainsi emploie ┌ excederoit le produit net d'un capital emploie ┌ dans l'emploi qui y approcha └t le plus pres dans l'ordre de productivite ┌ quant aux mines d'or et d'argent il ne s'ensuit pas parce que le metal est de cette nature la valeur du produit net d'une mine qui le renfermant est plus grande que celle du produit net de quelqu'autre metal que ce soit par exemple de fer ou de cuivre tout roule sur le rapport d'entre le capital emploie ┌ et le tant pour cent par an qu'on en retire

Figure 4-4, topic for French words



4076.tei
for topic 64: {thy, er, love, ev, ry, thee}

Keywords found for topic (marked pink in the text; off-topic keywords are shown in blue): heart, love, soul, thou, thy, tis, vain, ye

the watry glade where grateful science still adores her henry's holy shade and ye that from the stately brow of windsor's heights th expanse below of grove of lawn of mead survey whose turf whose shade whose flowers among wanders the hoary thames along his silver-winding way ah happy hills ah pleasing shade ah fields belov'd in vain where once my careless childhood stray'd a stranger yet to pain i feel the gales that from ye blow a momentary bliss bestow as waving fresh their gladsome wing my weary soul they seem to sooth and redolent of joy and youth to breathe a second spring say father thames for thou hast seen full many a sprightly race disporting on thy margent green the paths of pleasure trace who foremost now delight to cleave with pliant arm thy glassy wave the captive linnet which enthrall what idle progeny succeed to chase the rolling circle's speed or urge

Figure 4-5, Topic for lofty archaizing language in ECCO

4.3.1 Evaluation of the LDA-based approach

Possible scenario's relevant to our setting are:

1) Using unigram LDA models: perplexity experiments, where the document topic mix is either estimated from the in-domain training data or from first round recognition results.
2) SRILM has support for adaptation of an N-gram model to a unigram distribution. This could be used as a technically simple to obtaintopic-adapted N-gram models (adapt a generic N-gram model to a unigram LDA model specialized to a topic mix φ)

5 Category-based models

In TRANSCRIPTORIUM we have studied the use of category-based language models [Niesler & Woodland 1996] for document collections with a very regular structure, such as marriage register books. This language models strictly account for the syntactic structure of the documents, restricting the search space. In addition, category based language models have several advantages that can improve HTR system accuracy. For example, given that they share statistics between words of the same category, category-based models are able to generalise to word patterns never encountered in the training corpus. In addition, grouping words into categories can reduce the number of contexts in an n -gram model, and thereby reduce the training set sparseness problem.
On the other hand, these type of documents contain relevant semantic information that is very interesting for migratory studies, population research and genealogical investigation. Therefore, one of the goals of this kind of documents, is to extract this relevant information to allow the users to make use of it through semantic searches. The category-based language models studied here can be very useful for identifying and obtaining the relevant semantic information. For example, in the case of marriage register books, the husband's name, his job, the wife's name, etc.
In this section, we present the work carried out using a category-based language model for both automatic transcription and for relevant semantic information detection of marriage register books. This work has been presented in [Romero & Sanchez 2013].

5.1 Formal framework

Formally speaking, let $C = \{c_1, c_2, ..., c_N\}$ be a set of categories, each word of the vocabulary can belong to more than one category. Therefore, we define $C(w_i)$ as the set of all categories to which the word w_i can belong. Now assuming that the probability of a word w_i only depends of the category to which it belongs, and following the discussion presented in [Niesler & Woodland 1996], the probability of a word sequence $w = \{w_1 w_2 ... w_l\}$ using a category n -gram language model is computed as:

$$p(w) = \prod_{i=1}^{l} \sum_{c_i \in C(w_i)} p(w_i \mid c_i) \cdot p(c_i \mid c_{i-1} c_{i-2} ... c_{i-n+1})$$

Approximating the sum by the dominating term we can rewrite equations 1 as:

$$p(w) \approx \prod_{i=1}^{l} \max_{c_i \in C(w_i)} p(w_i \mid c_i) \cdot p(c_i \mid c_{i-1} c_{i-2} ... c_{i-n+1})$$

The parameters of the distributions of words into categories were calculated as:

$$p(w_i \mid c) = \frac{N(w_i \mid c)}{\sum_{w'} N(w' \mid c)}$$

where $N(w_i \mid c)$ is the number of times the word w_i has been labelled by c in the training corpus. Finally, as explained in [Nevado et. al. 2001], we can obtain the semantic information of a sentence from the semantic categories used in the most probable derivation for that sentence. We can define the semantic decoding $\hat{D} = c_1 ... c_l$ of a sentence $w_1 ... w_l$ as:

$$\hat{D} \approx argmax_D \prod_{i=1}^{l} p(w_i \mid c_i) \cdot p(c_i \mid c_{i-1} c_{i-2} ... c_{i-n+1})$$

6 Experiments

6.1 *Combining in-domain and out-of-domain training material: Experiments with the Bentham Data*

In this section we set up several experiments to show the effect of the proposed methods for domain adaption in HTR system. We first perform the base line experiments using optimized settings to build language modeling and then show the effect of these tuning parameters of the LM on the performance of the HTR system. We then perform a set of experiments for the proposed algorithms for sample selection to domain adaptation, i.e. the *Agree-Co* and *Disagree-Co* algorithms. We further make several combination strategies to interpolate the resulting language models. We finally report the results of HTR system using the proposed LMs and the base line methods.

In the experiments, we use three resources for training (cf. section 3.1: Bentham "Batch 1", Bentham without Batch 1, ECCO) and a separate test set for evaluating the HTR system.

6.1.1 LM Interpolation experiments

Our first set of experiments is about findingan optimal way to combine resources by language model interpolation. The relevant tools are the*InterpolateTwoLMS.sh* and *InterpolateThreeLMS.sh* scripts (cf.

Appendix A), which determine and apply optimal parameters for the interpolation of LMs trained from the Bentham In and Out domain and ECCO resources. As already stated, these methods are the baseline for the more advances methods.
We have applied the following scenarios:

1. Combining two Bentham resources (In and Out) and using a dictionary from the merged data to train the LM (Merged-InOut-Dic-InOut).
2. Interpolating Bentham In and Out domain resources using dictionary from In domain data (Inter-InOut-Dic-In).
3. Interpolating Bentham In and Out domain resources using dictionary from both In and Out domain data (Inter-InOut-Dic-InOut).
4. Combining two Bentham resources and interpolate the resulting LM with the LM of ECCO using dictionary from the merged data to train the LM (Inter-In+OutECCO-Dic-InOutECCO).
5. Interpolating Bentham In and Out domain resources with ECCO collection using dictionary from Bentham In and Out domain data (Inter-InOutECCO-Dic-InOut).
6. Interpolating Bentham In and Out domain resources with ECCO collection using dictionary from all of them (Inter-InOutECCO-Dic-InOutECCO).

6.1.2 Sample selection experiments

The following scenarios are used for the proposed iterative algorithms for sample selection, *Agree-Co* and *Disagree-Co*:

1) Single Iteration: In this scenario, a single iteration of the algorithm is used to select a set of high confidence resources. We select the best 15% of the high confidence resources using the proposed algorithms. The selected set is then used to build a LM. Next the three LMs (Bentham-in, Bentham-out and the model trained on the result of the selection procedure) can be interpolated using *InterpolateThreeLMS package*. We call these experiments *Agree-Co-Single* and *Disagree-Co-Single*.
2) Multiple Iterations: In this scenario, we perform20 iterations. The selected ECCO resources can be used in two ways:
   a) As a separate set: We train a LM from this set, and interpolate it with the two Bentham LM's. We call these experiments*Agree-Co-Inter-Three*and *Disagree-Inter-Three*.
   b) Add to the original Bentham resources and interpolate the resulting two LMs:*Agree-Co-Inter-Two* and *Disagree-Inter-Two*.

We repeat these experiments for each of the proposed three ranking metrics. The results will be reported in the following section.

6.2 Experiments with the category-based model on the ESPOSALLES database

6.2.1.1 The ESPOSALLES database

In this database, the relevant information consisted in the semantic information that each content word conveyed. Each marriage license typically contains information about the marriage day, husband's and wife's names, the husband's occupation, the husband's and wife's former marital status, and the socio-economic position given by the amount of the fee. In some cases, additional information is given as well, viz. the father's names and their occupations, information about a deceased parent, place of residence, or geographical origin. Taking into account the relevant semantic information of the licenses, 50 different categories were defined. This categories were defined by demographer experts. The relevant words in each license were manually labelled with the corresponding categories. In the following text we can see the annotated license corresponding to the image in Figure 1, where each semantic label is intermediately after the relevant word:

Dit dia rebere$ de Raphel [Husband's Name] Joani [Husband's Surname] texidor_de_lli [Husband's Profession] de Vilassar [Husband's Residence] fill de Miquel [Name of Husband's Father] Joani [Surname of Husband's Father] texidor_de_lli [Profession of Husband's Father] y de Violant [Name of Husband's Mother], ab Sperensa [Wife's Name] do$sella filla de Sebastia_Garau [Name of Wife's Father] Pere [Surname of Wife's Father] Boter [Profession of Wife's Father] de dita_parrochia [Residence of Wife's Father] y de t. [Name of the Wife's Mother]

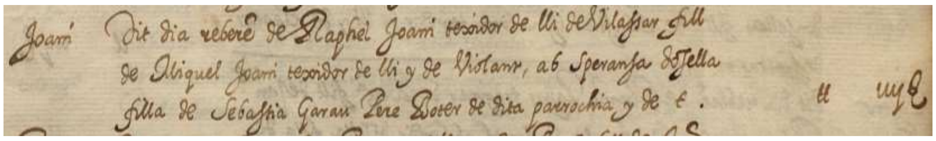

Joani Dit dia rebere de Raphel Joani texidor de lli de Vilassar fill
de Miquel Joani texidor de lli y de Violant, ab Speransa dosella
filla de Sebastia Garau Pere Boter de dita parrochia y de t. ll uyE

Figure 6-1, Example of a marriage license

Note that in some categories, the information was implicit in the license. For example, dita parrochia means “the mentioned parish” in English, that in turn refers to the town that is mentioned previously, that is Vilassar.
Given that, only some words of each license had relevant semantic information, the proposed categorisation involved classifying only some words in the vocabulary and not all of them. In this way, a partially categorised corpus was obtained, that is, not each word had a category associated to it.

6.2.1.2 Experimental setup

To carry out the experiments we have used the standard partition described in 3.2, consisting of seven consecutive blocks of 25 pages. The seven partitions have been used for cross-validation. That is, we carried out seven rounds, with each of the partitions used once as test data and the remaining six partitions used as training data.
In each round, the transcriptions of the training set were used to train a category-based bi-gram language model with Kneser-Ney back-off smoothing [Kneser & Ney 1995]. Similarly, the word category distributions were computed using Eq. 3 using only the transcription of the training set. This category-based language model was integrated into the handwritten text recognition process and helped us to extract the relevant semantic information of the different words and also to discover relationships between categories. The out of vocabulary (OOV) words (words of the test partition that do not appear in the training partition) belonging to a category seen in training are added as singletons to the corresponding word category distribution. For OOV words that belong to a category that has not been seen in training, we add the category in the category-based 1-gram and the word in the category distribution as singleton.

As previously explained, only some words of each license had relevant semantic information, resulting in a partially categorised corpus. Words that had not a category are viewed as categories that contain a single word. For instance, we introduced the category “DIA” containing only the word “dia”. For that categories that only have one word, the probability of that word into the category is 1. Following with the previous example $p(dia \mid DIA) = 1.0$. On the other hand, it is also important to note that, a word may belong to several categories. For example, the word *Ferrer* (that could be translated as Smith) could belong to the categories *husband surname*, *husband profession*, *father husband surname*, *father husband profession*, *wife surname*, *father wife surname*, etc.

6.2.1.3 Assessment measures

Different evaluation measures were adopted to assess the HTR and the relevant information extraction performance. The quality of the transcription is given by the well known *Word Error Rate* (WER).

On the other hand, to asses the quality of the information extraction performance we have used the standard precision and recall measures. We define precision and recall in terms of the number of relevant words in the dataset and the number of relevant word retrieved by the system. Relevant words are those words that belong to one of the 50 relevant categories defined by demographer experts. For instance, in the example shown in the previous subsection, the relevant words are those associated to a category: *Raphel, Joani, texidor_de_lli, Vilassar, Miquel, ...* Let R be the number of relevant words contained in the document, let D be the number of relevant words that the system has detected, and let C be the number of the relevant words correctly detected by the system. Precision (π) and recall (ρ) are computed as:

$$\pi = \frac{C}{D} \qquad \rho = \frac{C}{R}.$$

7 Results

7.1 Comparing baseline methods with intelligent sampling methods on the Bentham collection

In this section we report the results of the recognition experiments with the HTR system. Table 2 shows the performance of the HTR system based on the proposed baseline methods and algorithms in terms of the word error rate (WER). We have considered three main criteria for each experiment, the general word error rate, the word error rate removing the first word of each line[14], and the character error rate (CER)[15].

[14]This number is of interest because it emphasizes the importance of dealing with the line transitions, both in language modeling and in the core recognition engine, cf. section 1.3.2.

[15]I (Jesse) cannot collect all the data (notably for the *Disagree-Co* experiments) from home, so CER information is not yet complete

Table 7-1, The results of the baseline methods for HTR system

Method	WER %	WER without first word %	CER %	OOV %
Initial model using only Batch 1 training set[16]	34.5	34.3	19.9	9.44
Merged-InOut-Dic-InOut	34.01	-	-	-
Inter-InOut-Dic-In	33.40	-	-	-
Inter-InOut-Dic-InOut	30.02	24.57	-	-
Inter-In+OutECCO-Dic-InOutECCO	31.7	26	16.5	-
Inter-InOutECCO-Dic-InOut	30.7	25.3	15.9	-
Inter-InOutECCO-Dic-InOutECCO	**28.31**	**22.74**	**14.7**	**5.4**

Table 7-1 shows that interpolating LM of In domain data with the resulting LMs from out domain and ECCO resources improves the performance of the HTR system. In other words, these results emphasize that the out-of-domain data contains useful information. It is interesting to see, though, that the character error rate does not improve along with the word error rate. This may be due to the fact that model interpolation parameters are selected by perplexity, which is a word-level criterion. We also give the OOV percentage for the initial and the final model.

Tables 7-2 to 7-4 show the performance of the HTR system using the *Agree-Co* algorithm for domain adaptation with different selection metrics. As can be seen, the *Agree-Co* algorithm performs better than the baseline methods in most of the cases.

Table 7-2, The results of HTR system using the Agree-Co *algorithm, with the additive criterion (cf. 1.5.1) as selection metric.*

Method	WER %	WER without first word %	CER %
Agree-Co-Single	27.47	22.06	14.4
Agree-Co-Inter-Two	30.44	24.57	16.1
Agree-Co-Inter-Three	**27.13**	**21.68**	**14.2**

Table 7-3, The results of HTR system using the Agree-Co*, with the multiplicative criterion (cf. 1.5.1) as selection metric.*

Method	WER %	WER without first word %	CER %
Agree-Co-Single	**27.04**	**21.68**	**14.1**
Agree-Co-Inter-Two	28.74	23.22	15.4
Agree-Co-Inter-Three	27.30	21.97	14.3

[16] This result is different from the one reported in D3.1.2 (29% WER), which can be explained by our using an earlier version of the HTR package in these experiments

Table 7-4, The results of HTR system using the Agree-Co algorithm, with average word probability as selection metric (cf. 1.5.1)

Method	WER %	WER without first word %	CER %
Agree-Co-Single	28.02	23.12	14.9
Agree-Co-Inter-Two	29.51	23.99	-
Agree-Co-Inter-Three	**27.72**	**22.35**	**14.6**

Tables 7-5 and 7-6 show the performance of the HTR system using the *Disagree-Co* algorithm for sample selection. As can be seen in table 6, this approach is also effective and can improve the performance of the HTR system and outperforms the base line methods.

Table 7-5, The result of the HTR system with the Disagree-Co *algorithm and the multiplicative criterion*

Method	WER %	WER without first word %	CER %
Disagree-Co-Single	**26.95**	**21.58**	**14.1**
Disagree-Co-Inter-Two	28.75	23.03	-
Disagree-Co-Inter-Three	27.13	21.68	14.3

*Table 7-6, The results of HTR system with the*Disagree-Co *algorithm and the average word probability criterion*

Method	WER %	WER without first word %	CER %
Disagree-Co-Single	28.01	22.78	-
Disagree-Co-Inter-Two	28.99	23.31	-
Disagree-Co-Inter-Three	**27.63**	**22.16**	**14.5**

7.2 Results of category-based modeling on the Esposalles database

Experiments were carried out to test the performance of a category-based language model in both the handwriting recognition and the semantic information extraction process of a marriage license book. As explained in [Romero et. al. 2013], the ESPOSALLES database is provided at two different levels: line level and license level. In this work we have carried out experiments at license level. Given that the lines belonging to each license was known, feature sequences extracted from the lines could be easily merged into whole license line images.

Table 7-7 presents the experimental results obtained. Our baseline system was a HMM-based HTR system using a conventional 2-gram word language model (W-HTR). It was the same used in Deliverable D3.1.2 [TRANSCRIPTORIUM D3.1.2]. From the results we can see that a reduction in the value of WER was obtained when semantic categories were used. Most specifically, using the word-based language model (W-HTR) a WER of 11% was obtained, whereas for the category-based language model, the obtained WER decreased to 10.1%.

Table 7-7, Word Error Rate (WER),Character Error Rate (CER), precision (π), and recall (ρ)obtained with the a conventional word-based HTR system (W-HTR) and with the category-based HTR system (CB-HTR). All results are percentages

	WER	CER	π	ρ
W-HTR	11.0%	5.7%	-	-
CB-HTR	10.1%	5.6%	69,1	69.2

On the other hand, with respect to the information extraction performance, it is important to note that, using the bi-gram word language model we can not automatically extract any relevant word. In this case, the user should extract manually the relevant words and then move them to the appropriate structure/database. However, using the category-based language model, for each transcription hypothesis we can keep the relevant words related with the most probable semantic decoding, allowing the system to fill out some of the fields of the database automatically. The recall obtained using the category based language model is 69.2%, which means that the 69.2% of the 21,336 relevant words in the database have been correctly extracted. On the other hand, the precision obtained means that, from all the relevant word extracted by the system, the 69.1% were correct.

8 Conclusions

We have studied and tested several ways in which task-specific approaches to language modeling can improve handwritten text recognition results, when the resulting language models are deployed in the TRANSCRIPTORIUM HTR system.
On the one hand, approaches to the combination of in-domain and out-of domain data have been shown to yield improvement in HTR results, both using established techniques (model interpolation) and novel approaches based on intelligent sample selection in a co-training inspired setting. These approaches have been tested on the Bentham dataset, with the*Eighteenth Century Collections Online* Corpus as out-of-domain data.The experimental results show that our proposed approach for domain adaptation can effectively exploit informative data from the out of domain data and improve significantly the recognition performance of the HTR system.
On the other hand, we have studied the use of a category-based language model to relevant information extraction and for automatically transcription of a marriage license book. These documents were used for centuries by ecclesiastical institutions to register marriages. They have interesting information used by demographers, who devote a lot of time to extract the relevant information and to transcribe them. Given the interesting semantic information included in the licenses, we have used it to define the categories used in the category-based language model. From the results we can see that this category based language model can be useful to automatically extract the relevant information, helping the user in this hard task. In addition, using this language model during the transcription process improves the HTR system accuracy with respect to using a conventional n-gram language model.

9 Future work

In 2014, we plan to extend the work on sample selection for different datasets and combine it with elaboration of the approach to text normalization for historical documents, leveraging the historical lexical data acquired by, and specifically developed in the project,starting from the late medieval Dutch data. The implementation of the topic-modeling approach will be finishedand put to the test.

Another direction for the future work is to use a clustering based approach in order to find similar data points to in-domain data and find an iterative way to combine more similar clusters.
It may be desirable to revisit the way in which the HTR engine interacts with the language models. As has been shown in the literature, approaches such as class-based [Brown et. al. 1992] and neural-network based [Bengio et. al. 2006] models can improve the performance of the language modeling component of an NLP system. In order to apply such approaches, we need to modify the current system setup in which language models are loaded from a file in HTK lattice format, which restricts the class of language models that can be used to classical N-gram models.

10 References

Gilles Adda , Martine Adda-decker , Jean-luc Gauvain, Lori Lamel. Text Normalization And Speech Recognition In French, *Proc. ESCA Eurospeech'97*, 1997.

Adda-decker, M., Adda, G., Gauvain, J., Lamel, L.. Large vocabulary speech recognition in French,*Proceedings of IEEE International Conference on of Acoustics, Speech, and Signal Processing*, 1999.

Bellegarda, J. R. (2004). Statistical language model adaptation: review and perspectives. *Speech Communication*, *42*(1), 93–108. doi:10.1016/j.specom.2003.08.002

Yoshua Bengio, Holger Schwenk, Jean-Sébastien Senécal, Fréderic Morin, Jean-Luc Gauvain. Neural Probabilistic Language Models. *Studies in Fuzziness and Soft Computing* Volume 194, 2006, pp 137-186. doi:10.1007/3-540-33486-6_6

J. Bilmes and K. Kirchhoff. "Factored Language Models and Generalized Parallel Backoff". *Human Language Technology Conference, 2003.*

Blei, David M.; Ng, Andrew Y.; Jordan, Michael I, Latent Dirichlet allocation. In Lafferty, John. *Journal of Machine Learning Research*3 (4–5): pp. 993–1022, 2003. doi:10.1162/jmlr.2003.3.4-5.993.

Peter F. Brown, Vincent J. Della Pietra, Peter V. deSouza, Jennifer C. Lai and Robert L. Mercer. Class-Based n-gram Models of Natural Language. *Computational Linguistics* 18(4) (1992):pp. 467-479.

Victor Chahuneau, Noah A. Smith, and Chris Dyer. Knowledge-Rich Morphological Priors for Bayesian Language Models. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics* (NAACL 2013), Atlanta, GA, June 2013.

Stanley F. Chen, Douglas Beeferman, and Ronald Rosenfeld. Evaluation metrics for language models. In *DARPA Broadcast News Transcription and Understanding Workshop*, 1998.

Jorge Civera, Alfons Juan. Domain Adaptation in Statistical Machine Translation with Mixture Modelling. *Proceedings of the Second Workshop on Statistical Machine Translation*, 2007. pp. 177-180. Association for Computational Linguistics.

J. Gao, M. Li, and K.F. Lee. N-gram distribution based language model adaptation. In *Proc. of the ICSLP*, pages 16-20, Beijing, 2000.

Guillem Gascó, Martha-Alicia Rocha, Germán Sanchis-Trilles, Jesús Andrés-Ferrer, Francisco Casacuberta. *Does more data always yield better translations*?. Proceedings of the 13th European Chapter of the Association for Computational Linguistics, 2012. pp. 152-161.

Kestemont, M., W. Daelemans & G. De Pauw, ‘Weigh your words: Memory-based lemma-retrieval for Middle Dutch literary texts’. Paper presented at *CLIN 2010. Computational linguistics in the Netherlands 20*. Utrecht University, 2010.

Klakow, D., & Peters, J. Testing the correlation of word error rate and perplexity. *Speech Communication*, *38*(1-2), 19–28, 2002 doi:10.1016/S0167-6393(01)00041-3

R. Kneser and H. Ney. Improved backing-off for m-gram language modeling. volume 1, pages 181–184, Detroit, USA, 1995.

Lavrenko, Victor, Toni M. Rath, and R. Manmatha. Holistic word recognition for handwritten historical documents. In *Document Image Analysis for Libraries, 2004. Proceedings. First*

International Workshop on, pp. 278-287. IEEE, 2004.

Lindsey, R. V, Iii, W. P. H., & Stipicevic, M. J. A Phrase-Discovering Topic Model Using Hierarchical Pitman-Yor Processes, *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pp. 214–222, 2012.

Malagon, C. ; Rizky, R. ; Kim, Y. ; Marzal, F. ; Izquierdo, L. Automatic Abbreviation Detection in Medieval Medical Documents. *International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2012, pp. 337 – 342.

Christof Monz. Statistical Machine Translation with Local Language Models. In *Proceedings of Emperical Methods for Natural Language Processing (EMNLP 2011)*, pages 869-879, 2011.

R.C. Moore and W. Lewis. Intelligent selection of language model training data. In *Proc. of the ACL 2010 Conference Short Papers*, pages 220-224, July 2010.

Francisco J. Nevado, Joan-Andreu Sánchez, and José-Miguel Benedí. Lexical decoding based on the combination of category-based stochastic models and word-category distribution models. In *IX Spanish Symposium on Pattern Recognition and Image Analysis*, pages 183–188. Publicacions de la Universitat Jaume I, 2001.

T.R. Niesler and P.C. Woodland. A variable-length category-based n-gram language model. In *Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on*, volume 1, pages 164 –167 vol. 1, may 1996.

K. Nigam and R. Ghani, “Analyzing the effectiveness and applicability of co-training,” in CIKM . ACM, 2000, pp. 86–93.

Verónica Romero, Joan-Andreu Sánchez, Nicolás Serrano, and Enrique Vidal. Handwritten text recognition for marriage register books. In *Proc. of the 11th International Conference on Document Analysis and Recognition (ICDAR 2011)*, pages 533–537, Beijing, China, 2011.

V. Romero, A. Fornés, N. Serrano, J. A. Sánchez, A.H. Toselli, V. Frinken, E. Vidal, and J. Lladós. The ESPOSALLES database: An ancient marriage license corpus for off-line handwriting recognition. *Pattern Recognition*, 46:1658–1669, 2013. http://dx.doi.org/10.1016/j.patcog.2012.11.024.

V. Romero and J. A. Sánchez. Category-based language models for handwriting recognition of marriage license books. In *International Conference on Document Analysis and Recognition*, pages 788–792, 2013.

Scharenborg, Odette, Stephanie Seneff, and Lou Boves. A two-pass approach for handling out-of-vocabulary words in a large vocabulary recognition task. *Computer Speech & Language* 21, no. 1 (2007): 206-218.

Stolcke, Andreas, et al. "SRILM at sixteen: Update and outlook." *Proceedings of IEEE Automatic Speech Recognition and Understanding Workshop*. 2011.

Tanha Jafar, Ensemble approaches to semi-supervised learning, Ph.D. thesis, University of Amsterdam, 2013.

Tanha, Jafar, Maarten van Someren, and Hamideh Afsarmanesh. Disagreement-based co-training. In *Tools with Artificial Intelligence (ICTAI), 2011 23rd IEEE International Conference on*, pp. 803-810. IEEE, 2011.

TRANSCRIPTORIUM. D.3.1.2. Description and evaluation of tools for DIA, HTR and KWS. 2014.

Joerg Ueberla, Analysing a simple language model - some general conclusions for language models for speech recognition, *Computer Speech & Language* Volume 8, Issue 2, April 1994, pp. 153–176.

Wallach, B. H. M. Topic Modelling□: Beyond Bag of Words. *Proceedings of the 23rdInternational Conference on Machine Learning*, Pittsburgh, PA, 2006.

Wang, X., McCallum, A., & Wei, X. (2007). Topical N-Grams: Phrase and Topic Discovery, with an Application to Information Retrieval. *Seventh IEEE International Conference on Data Mining (ICDM 2007)*, 697–702. doi:10.1109/ICDM.2007.86

Qiu-Feng Wang, Fei Yin, Cheng-Lin Liu, Unsupervised language model adaptation for handwritten Chinese text recognition, *Pattern Recognition* Volume 47, Issue 3, March 2014, Pages 1202–1216 (DOI: http://dx.doi.org/10.1016/j.patcog.2013.09.015).

Wuthrich, M. ; Liwicki, M. ; Fischer, A. ; Indermuhle, E. ; Bunke, H. ; Viehhauser, G. ; Stolz, M. Language Model Integration for the Recognition of Handwritten Medieval Documents, *ICDAR '09*. (10th International Conference on *Document Analysis and Recognition*), pp.211 – 215, 2009.

11 Appendix: TranScriptorium tools for language modeling

The language modeling relies on an up-to-date installation of the SRILM toolkit.

11.1 Cleaning Text

This tool serves to remove characters which are not recognized by the HTR system from the text before further processing. It also escapes characters which create problems for the HTK tools.
To apply it, create a plain text file with the list of accepted characters (the *CharacterSet.txt* file) and copy this file to the folder where the cleaned text should be created. The command line is as follows:

```
java –jar WProc.jar Inputfile CharacterSet OutputFile.
```

11.2 Normalizing Text

In order to build a language model, we need to apply text normalization. The developed program for normalization can be used as follows:

```
java –jar WNorm.jar CleanedText  OutputFile.
```

11.3 Building a word frequency list

The word lists are used in subsequent steps to construct the HTR dictionary.
This program can be used to build the word list and word frequency list as follows:

```
java –jar WList.jar –i filename –o filename –n number –s filename
```

The command line options:

-i input file name
-o output file name, which is the word list
-n cutoff point in terms of the number of frequency
-s output file name, containing the sorted word frequency list

11.4 Making a Dictionary

For use with the HTR system, and in order to generate a HTK lattice file from an ARPA-style language model file, we need a HTR dictionary in HTK format. This program generates a dictionary which includes a relative probability for each realization of a normalized word form in recognizable characters:

```
"HUMBLE"    [Humble]        .071078431      H u m b l e @
"HUMBLE"    [humble]        .924019608      h u m b l e @
```

This program can be used as follows:

```
java –jar  WDic.jar  wordlistfile frequencylistfile outputfilename
```

The input should the word list and word frequency files that have been generated by the WList.jar program.

11.5 Training N-gram language models

To train the N-gram LM, we use the *make-big-lm* script of SRILM [Kneser & Ney 1995]. It constructs large N-gram models in a more memory-efficient way than *ngram-count*scriptby itself. This script first loads N-gram counts files and generates the N-gram model using different smoothing algorithms. Figure 1 (cf. section 2.2) depicts a workflow to construct a N-gram LM from a corpus.
The main script for producing N-gram LM is called as follows:

```
bash TrainingLM.sh inputfile CharacterSet OutputFolder Cutoff
```

where *OutputFolder* is used to record all outputs and *Cutoff* is used to select the words with frequency more than *Cutoff* point.
In order to use this script, the user will have to modify the settings for the SRILM_HOME and PATH variables according to his system setup.

11.6 Evaluation of N-gram models

This script evaluates an N-gram LM model in ARPA format on a test set. The command line:

```
bash TestingLM.sh TestSet LMFolder OutputFile
```

The output of this script is a file containing the perplexity of the test set per sentence and the general text perplexity in the end of the generated output file.
In order to use this script, the user will have to modify the settings for the SRILM_HOME and PATH variables according to his system setup.

11.7 Making Lattice File

The following script produces the required files in HTK format, for use in the HTR system.

bash LatticeFromLM.sh *OriginalTextResource CharacterSet OutPutFolder CutoffPoint*.

The output will be the *latticeFile.txt* in the *OutPutFolder*
Again, please adapt the internal settings for PATH and SRILM_HOME.

11.8 Interpolating Language Models

We use the *compute-best-mix* script of SRILM to find the optimal weights for two LMs in the *ComputeweightsTwoLMS.sh* script, with the following command line:

```
bash ComputeweightsTwoLMS.sh PPLFile1 PPLFile2
```

This produces the best value (in terms of perplexity) for the interpolation parameter λ.

The script *InterpolateTwoLMS.sh*is uses the optimal $\lambda$to interpolate two LMs, and generatesthe required output format for the HTR system:

```
 bash InterpolateTwoLMS.sh FirstLM SecondLM OutputFolder Lambda CutoffPoint TestSet
```

where *CutoffPoint* is the minimum word frequencyfor words to be included in the language model and the HTR dictionary, and and *TestSet* is used for evaluation of the resulting LM.

We also developed a script for interpolating three LMs:

```
       bash ComputeweightsThreeLMS.sh PPLFile1 PPLFile2 PPLFile3
```

the results of which are used to interpolate three LMs.

Here we show a part of the output as:

```
       iteration 12, lambda = (0.669546 0.316303 0.0141513), ppl = 68.6298
       iteration 13, lambda = (0.669257 0.318701 0.012042), ppl = 68.6079
       iteration 14, lambda = (0.668991 0.320619 0.0103898), ppl = 68.5923
       iteration 15, lambda = (0.66876 0.32215 0.00909002), ppl = 68.5812
       iteration 16, lambda = (0.668567 0.323369 0.00806402), ppl = 68.5734
       1271 non-oov words, best lambda (0.668408 0.32434 0.00725181)
```

The last line of the output shows the resulting weights, λ_1=0. ***668408***, λ_2=0. ***32434*** , λ_3=0. ***00725181.***
We then use λ_1, λ_2 and λ_3 in the following script:

```
bash InterpolateThreeLMS.sh FirstLM SecondLM ThirdLM OutputFolder lambda1 lambda2
lambda3  CutoffPoint TestSet
```

For all concerned scripts, please adapt the internal settings for PATH and SRILM_HOME.

11.9 Tools for intelligent data selection

It takes three steps to run the proposed iterative approaches for sample selection:

1. Clean the ECCO (out-of-domain) dataset
2. Run the *Agree-Co* or *Disagree-Co* algorithm for sample selection.
3. Use the selected samples for training a LM and applyingthe interpolation methods to combine in-domain and selected out-of-domain data

We have developed the tool *TClean.jar*for cleaning and normalizing the ECCO dataset, to be invoked as follows:

```
     Java –jar TClean.jar InputFolder CharacterSet.txt Output_CleanedTextFolder
            Output_NormalizedAndCleanedTextFolder RequiredProgramsFolder
```

where

- *InputFolder* is the folder of ECCO resources,
- *CharacterSet* is a text file that contains the applicable character set for preprocessing the text, as describedin Section 11.1
- *Output_CleanedTextFolder* is the folder where the cleaned text will be stored
- *Output_NormalizedAndCleanedTextFolder* is the folder for the normalized and cleaned resources.

- *RequiredProgramsFolder* is the location of the required .jar files for preprocessing the text, like WProc.jar and WNorm.jar.

For instance:

```
Java –jar TClean.jar /mnt/Projecten/TRANSCRIPTORIUM/Data/Corpora/ECCO/LPL_For_Each_Doc/
/mnt/Projecten/TRANSCRIPTORIUM/Tools/languagemodeling/CharacterSet.txt
/mnt/Projecten/TRANSCRIPTORIUM/Tools/languagemodeling/TestSampleSelection/CleanedTxt
/mnt/Projecten/TRANSCRIPTORIUM/Tools/languagemodeling/TestSampleSelection/CleanedAndNormalized/mn
t/Projecten/TRANSCRIPTORIUM/Tools/languagemodeling/.
```

Now the ECCO text is ready to use. The second step is to apply the proposed algorithms for sample selection. As we mentioned in Section 4, there are two collections of Bentham, i.e. In-domain and Out-of-domain Bentham. We use here the cleaned and normalized text of these two collections and call them *cleanedText.txt* and *normalizedText.txt*. The tool *AgreeCo.jar*which implements the proposed *Agree-Co*has the following command line syntax:

```
    Java -jar AgreeCo.jar SelectionPercent NumberIterations Confidence InDomainF
           OutDomainF ECCONormalizedDomainF ECCOCleanedDomainF OutputInF
     OutputOutFNameofSelectedSamples.txt TypeOfCriterion Requiredprogramfolder
```

where

- *SelectionPercent* is the percentage that the algorithm selects at each iteration,
- *NumberIterations* is the number of Iterations.,
- *Confidence* is a threshold for confidence (after the first iteration the confidence measure can be estimated properly)
- *InDomainF* is the location(folder) of In-domain LM and Data
- *OutDomainF* is the location of Out-of-domain LM and Data,
- *ECCONormalizedDomainF* is the location of ECCO Normalized data,
- *ECCOCleanedDomainF*is the location of ECCO Cleaned data,
- *OutputInF* is the location where the in-domain LM is saved
- *OutputOutF* is the location for the out-of-domain LM,
- *NameofSelectedSamples* is the resulting sample selection file
- *TypeOfCriterion*: Select the ranking criterion, possible values are
  - 1: for adding PPL + OOVs,
  - 2: PPL*OOVs,
  - 3: (1/PPL)*(1-OOVs)
- *Requiredprogramfolder* is the folder of required .sh files for LM building, like testPPLJava.sh trainingLMJava.sh

An example call of the TClean.jar tool:

```
AgreeCo.jar 1 102
/mnt/Projecten/TRANSCRIPTORIUM/Tools/languagemodeling/TestSampleSelection/InBentham/
/mnt/Projecten/TRANSCRIPTORIUM/Tools/languagemodeling/TestSampleSelection/OutBentham/
/mnt/Projecten/TRANSCRIPTORIUM/Tools/languagemodeling/TestSampleSelection/CleanedAndNor
malized/
/mnt/Projecten/TRANSCRIPTORIUM/Tools/languagemodeling/TestSampleSelection/CleanedText/
/mnt/Projecten/TRANSCRIPTORIUM/Tools/languagemodeling/TestSampleSelection/In/
/mnt/Projecten/TRANSCRIPTORIUM/Tools/languagemodeling/TestSampleSelection/Out/
/mnt/Projecten/TRANSCRIPTORIUM/Tools/languagemodeling/TestSampleSelection/TestSampledSe
lection.txt 1 /mnt/Projecten/TRANSCRIPTORIUM/Tools/languagemodeling/
```

We now have the selected resources from the out-of-domain (ECCO) data. We can thenapply different scenarios to combine resources by language model interpolation[17]:

- Interpolate $LM_{sampled\text{-}data}$with $LM_{Bentham\text{-}In}$ and $LM_{Bentham\text{-}Out}$
- Interpolate $LM_{sampleddata+Bentham\text{-}In}$ with$LM_{Bentham\text{-}Out}$

Exactly the same scenario can be used to run the the tool *DisagreeCo.jar,*which implements the disagreement-based approach (section 4.2.2).

11.10 Topic modeling

```
[will be in next release, but we will manage to get some experiments
done before the ASM and the review meeting]
```

[17]Cf section 11.8 for details on the interpolation process. For more information about the experimental setup cf. Section 7.1.