Handwriting Recognition in Historical Documents using Very Large Vocabularies

Volkmar Frinken
Centro de Visio per Computador
Universidad Autonoma de Barcelona
Edifici O, 08193 Bellaterra, Barcelona, Spain
vfrinken@cvc.uab.es

Andreas Fischer
CENPARMI
Concordia University
Montreal, H3G 1M8, Canada
an_fisch@encs.concordia.ca

Carlos-D. Martínez-Hinarejos
Instituto Tecnológico de Informática
Universitat Politècnica de València
Camino de Vera, s/n, 46022 Valencia, Spain
cmartine@dsic.upv.es

ABSTRACT

Language models are used in automatic transcription system to resolve ambiguities. This is done by limiting the vocabulary of words that can be recognized as well as estimating the n-gram probability of the words in the given text. In the context of historical documents, a non-unified spelling and the limited amount of written text pose a substantial problem for the selection of the recognizable vocabulary as well as the computation of the word probabilities. In this paper we propose for the transcription of historical Spanish text to keep the corpus for the n-gram limited to a sample of the target text, but expand the vocabulary with words gathered from external resources. We analyze the performance of such a transcription system with different sizes of external vocabularies and demonstrate the applicability and the significant increase in recognition accuracy of using up to 300 thousand external words.

Keywords

Historical Documents, Handwriting Recognition, Language Modeling, Google N-grams, BLSTM Neural Networks

1. INTRODUCTION

In the context of preserving the humankind's cultural heritage, large efforts are being done to scan and store vast amounts of historical data. The digitization, however, is only the first step on the way to make the contents readily accessible to researchers as well as the general public. A major problem up to date is to extract the textual content from those images into a computer-readable format, which is tedious, time-consuming work and requires expert knowledge.

HIP '13 August 24 2013, Washington, DC, USA

The last years have seen increased research activities of document analysis for historical data [2, 3] and recent advances have made a (semi-)automatic processing a viable choice for accessing the contents through keyword spotting [7], interactive or full automatic transcription systems [6, 19].

However, automatic handwriting recognition is a difficult problem that is not yet solved. Some of the key problems are large varieties in handwriting styles and the need to understand contextual cues through adequate language modeling to resolve ambiguities. Both points are even harder for historic data. Limited amount of transcribed training data for a specific writing style, non-uniform spelling rules, frequent use of abbreviations as well as special symbols can usually be observed. In addition, given an historic text to transcribe, it is very likely that comparable language samples do not exist, as far as time, location, and context is concerned, all of which are important factors when modeling the text via external sources.

In this work we focus on the language modeling aspect and demonstrate a recognition system that uses limited, but accurate n-grams obtained from the training set of the handwriting recognition system and augment the language model with a very large vocabulary obtained from different sources. This maintains the language structure of the training set, which is expected to match the test data, while effectively reducing the out-of-vocabulary rate and significantly increasing the recognition rate.

A further contribution of this paper is the presentation of a working recognition system that can cope with very large vocabularies of several hundred thousand words, which is much more than existing systems [10, 14], to the knowledge of the authors.

The rest of this paper is structured as follows. In Section 2, the database on which we performed the study is introduced. Language modeling and considered corpora are discussed in Section 3 and the handwriting recognition system is explained in Section 4. Section 5 presents an experimental evaluation and conclusions are drawn in Section 6.

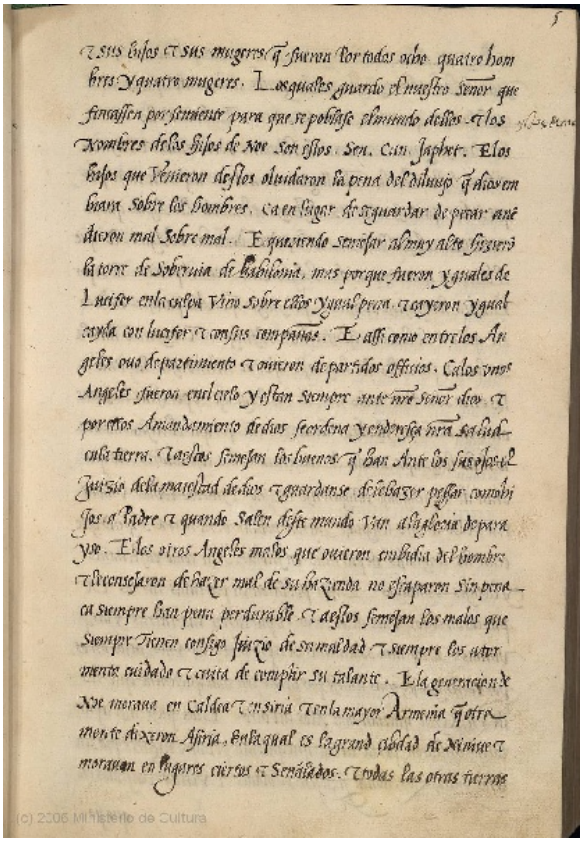

Figure 1: Typical page of the RODRIGO database.

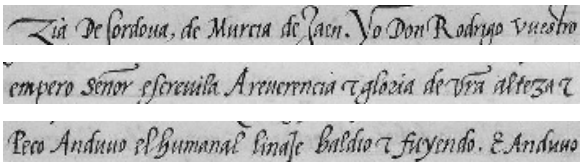

Figure 2: Examples of extracted lines from the RODRIGO database.

2. DATABASE

The database used in this work is the RODRIGO database [18], which corresponds to a single-writer Spanish text written in 1545, "Historia de España del arçobispo Don Rodrigo". The book has 853 pages with historical chronicles of Spain; most of the pages consist of a single block of well separated lines of calligraphical text. Image in Figure 1 shows an example of a typical page in this manuscript.

Lines in these pages were extracted and used as primary data. An example of lines obtained in this extraction process is shown in Figure 2. There is a total of 20 356 extracted lines in PNG format files. In the original database they are named after the page number and line number.

The set of lines was divided into three different sets: training (10 000 lines), validation (5 010 lines), and test (5 346 lines). The out-of-vocabulary rate of the test set is 6% given the vocabulary of the training and validation set.

3. LANGUAGE MODELING

It has long been known that external information about the target language can help resolve ambiguities and increase the recognition rate [16]. A common choice are statistical bi-gram models that contain a list of words as well as conditional occurrence probabilities $p(w|w')$ of a word w given the previous word w'. With this, the probability of a word sequence $\hat{w} = w_1 \cdots w_N$ can be approximated as

$$p(w_1)p(w_2|w_1)p(w_3|w_2)\cdots p(w_N|w_{N-1}) \quad .$$

This simple, yet powerful model can not grasp long-term dependencies between distant words, but it provides computational advantages, since it fulfills the Markov property.

The actual probabilities are not easily estimated [8]. Just counting the number of observations in a text overestimates rare words that appear by chance while other words that do not appear are assigned a probability of 0, which obviously is not correct. Additionally, a specific text is not a random sampling of words but deals with a certain topic. Hence, any general language model does not reflect the true probabilities and choosing the language corpus is therefore a challenging task. For historical data this problem is even more imminent since words, spelling variants, common abbreviations, and special symbols can change quickly over time and place can lead to a lack of independent data [20].

3.1 Measures

To help in the recognition process, a good language model should assign a high probability to likely sentences. This can be measured in terms of perplexity, which is a function of the average estimated word probability of a given text.

$$\text{ppl}(\hat{w}|LM) = 2^{-\frac{1}{N}\sum_{i=1}^{N}\log_2\left(p(w_i|w_0\cdots w_{i-1})\right)}$$

where LM is the language model and $\hat{w} = w_1 \cdots w_N$ a word sequence.[1] The lower the perplexity, the higher the average probability and therefore the predictive power of the model.

Note, however, that perplexities can not be easily compared when different underlying vocabularies are used for open vocabulary recognition task, since out-of-dictionary events in the test set are problematic. Usually, those words cannot be recognized, hence their probability would be 0 and the perplexity undefined. Assigning an arbitrary value to OOV words, in turn, does not reflect the transcription process.

Thus, the final recognition rate, given the same underlying recognizer, seems to be a more meaningful measure.

3.2 Google N-Grams

As a byproduct of the massive effort to scan and automatically transcribe millions of printed books, Google has gathered Tera-bytes of textual data and has published n-gram counts for $n = 1 \ldots 5$ for eight of the most wide-spoken languages [17]. Also, the n-gram counts are further subdivided into the year of the publication date of the corresponding book.

This makes the data an interesting external source for language information. The text of the database of our experi-

[1] w_0 is either a token indicating the start of the text or $p(w_1|w_0)$ is defined as $p(w_1)$

ments (see Section 2) is written in Old Spanish in the 16th century. Unfortunately, only three books from this period are scanned. Also, no manual correction was performed on the data, so that some OCR errors can be observed, i.e. historical long s (ſ) is often recorded as 'f'.

Thus, we considered a wider time frame to be relevant in order to create a large vocabulary and reduce the OOV rate in the recognition. High-order n-grams, however, did not improve the perplexity on the validation set in preliminary experiments, so this idea was not further perused.

We chose the year 1800 as a threshold to get a good trade-off between data quality and quantity. This subset consists of 9 388 out of the 854 649 books from the complete Spanish corpus, respectively a list of 1 361 298 unique words, sorted according to their frequency. Even though this is more manageable, 1.36 million words are still too much for most automatic recognition systems.

3.3 Don Quixote

Coincidentally, the famous Spanish masterpiece "Don Quixote" was written merely 60 years after the database and can therefore be expected to have at least a similar vocabulary.

An electronic edition of "Don Quixote" by Miguel de Cervantes Saavedra [1] that preserved the original spelling variants[2] and guaranteed to be free of OCR errors served therefore as a second language source. This edition contains 16 538 unique words all of which are added to the vocabulary.

4. BLSTM HANDWRITTEN TEXT RECOGNITION

The recognizer used in this work is based on bidirectional long short-term memory neural network [10]. The long short-term memory is a second-order recurrent neural network architecture, in which certain weights of the networks are given by the output of dedicated nodes. By controlling the input, output, and recurrence, a differential version of a memory cell can be simulated. This allows the network to access informations across several time-steps to cope with non-Markovian dependencies. The bidirectionally assures that context from both sides are considered.

The network classifies a sequence of input features into a sequence of posterior character probabilities which are then transformed into the most likely word sequence given a language model through a token passing algorithm. Instead of the original recognition algorithm [10], we rely on a more efficient token passing algorithm proposed in [4] which is able to cope with very large vocabularies, both in terms of memory and speed efficiency. Note that the bidirectionality is only used to compute the character probabilities. The token passing algorithm proceeds in the direction of writing from left to right.

[2]The famous introductory sentence "En un lugar de la Mancha de cuyo nombre no quiero acordarme, no ha mucho tiempo que vivía un hidalgo de los de lanza en astillero, ...", for example, used to be in the original "En vn lugar de la Mancha, de cuyo nombre no quiero acordarme, no ha mucho tiempo que viuia vn hidalgo de los de lança en astillero, ..."

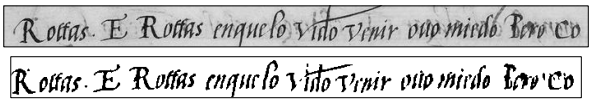

Figure 3: Text line preprocessing.

4.1 Preprocessing

The text line images of the RODRIGO database are preprocessed in three steps before recognition. First, the text foreground is extracted by means of binarization. Next, noise is removed from the binary image based on connected component analysis. Finally, the binarized images are represented by a sequence of feature vectors extracted with a sliding window.

For binarization, we follow the procedure proposed in [5]. First, edges are locally enhanced by means of Difference of Gaussians (DoG) using Gaussian kernels with radii $r_1 = 10.0$ and $r_2 = 0.5$, respectively. Afterwards, a global threshold is applied to the grayscale value of the pixels obtain a binary image. In this paper, we use a threshold of $0.75 \cdot T_0$ with respect to the Otsu threshold T_0. The parameters have been optimized by visual inspection on a few training samples. An exemplary result is illustrated in Figure 3 demonstrating the ability of the binarization method to deal with ink bleed-through in most cases.

In a next step, we remove noise from the binary images based on connected component (CC) analysis. First, pepper noise is dealt with by removing all CCs with less than 5 pixels. Then, text parts from the preceding text line are discarded by removing all CCs with a center of mass higher than 80% of the image height. To deal with touching text lines, large CCs connected to the top of the image are trimmed down to 80% of the image height. The effect can be seen in the example shown in Figure 3. Note that no skew or slant correction is applied.

Finally, a sliding window with a width of 1 pixel is moved from left to right over the binary image to extract a sequence of feature vectors $x_1, \ldots, x_N$ with $x_i \in \mathbb{R}^n$. At each of the N positions of the sliding window, $n = 9$ geometric features are extracted. Three global features include the fraction of black pixels, the center of gravity, and the second order moment. Six local features consist of the position of the upper and lower contour, the gradient of the upper and lower contour, the number of black-white transitions, and the fraction of black pixels between the contours. For more details on the geometric features, we refer to [15].

4.2 Recognition

Training of the BLSTM NN recognizer is done by iteratively adjusting randomly initialized weights via standard back-propagation through time [11]. The objective function is designed to minimize the negative log likelihood of the ground truth, given the network output [10].

Recognition is based on dynamic programming to obtain the most likely sequence of words for the feature vector sequence $x_1, \ldots, x_N$. Two quantities are optimized conjointly to find the best sequence of words. First, the posterior probabilities

of the morphological character models, which are calculated by the network for each feature vector, and secondly, language model probabilities in form of word bi-grams.

The original BLSTM NN recognition algorithm [10] is a token passing algorithm that performs the dynamic programming in a memory-efficient way. It loads all vocabulary words only once into the memory and performs the dynamic programming step by step from one sliding window position to the next. During each step, the algorithm iterates over all characters of all vocabulary words to update the partial recognition results. While this procedure guarantees an optimal solution, it is computationally challenging.

In order to cope with very large vocabularies with several hundred thousand words, we employ a more efficient token passing algorithm for BLSTM NN recognition, which was proposed in [4]. The algorithm has an improved memory and speed efficiency following two general strategies for handwriting recognition with large vocabularies [13]. First, a more compact representation of the vocabulary is used in form of a lexical tree. The lexical tree aggregates word prefixes that are shared by different words such that they are stored and processed only once. Secondly, a beam search is performed by pursuing only the n_1 best word endings at each each sliding window position and only the n_2 best partial recognition results overall. Although the pruned search is no longer optimal, we have observed no significant loss in accuracy on the validation set with the parameters $n_1 = 10$ and $n_2 = 15,000$ used in this paper.

For a detailed description of the token passing algorithm used for large-vocabulary recognition with BLSTM NN, we refer to [4].

5. EXPERIMENTAL EVALUATION

5.1 Setup

In order to evaluate the applicability and effectiveness of using very large vocabularies from external sources in the automatic transcription of historic handwritten text, we performed the following set of experiments. Neural networks are trained by initializing the weights with random values which are then changed in the learning step via back-propagation. Hence, for meaningful results, several networks can initialized with different weights. We trained 10 BLSTM neural networks on the training set using standard parameters with a learning rate of 10^{-4} and a momentum of 0.9. The number of LSTM notes in the hidden layer was set to 100. These are standard values that turned out to work well for a variety of handwritten data and were not further validated. The BLSTM NN implementation used in this paper is based on an earlier version of RNNLIB [9].

After training we selected the single best network according to the character error rate on the validation set. This network was then used to produce the matrices of output character probabilities for each of the lines in the test set. Keeping these fixed, the final transcriptions for the different language models were computed with the token passing algorithm from Section 4.2.

All language models use bi-gram probabilities estimated with modified Kneser-Ney smoothing [12] on the training and validation set and differ only in the list of known word. See Table 1 for a summary. An open vocabulary language model Int_{open} and a closed vocabulary language model Int_{closed} with additional words from the testing set are compared to seven language models with an additional external vocabulary between 20 thousand and 300 thousand words. The SRILM toolkit[3] is used for language model estimation.

Table 1: The list of the seven language models with external vocabulary and the two reference language models along with OOV rates and perplexities on the testing set. Note that the perplexities are only measured on the known words.

Name	ext. vocabulary	OOV rate	perplexity
Int_{open}	0	6.15	166.741
Ext_{20}	20k	4.70	192.854
Ext_{50}	50k	4.10	205.984
Ext_{100}	100k	3.59	219.128
Ext_{150}	150k	3.30	227.557
Ext_{200}	200k	3.11	233.606
Ext_{250}	250k	2.94	239.044
Ext_{300}	300k	2.80	243.824
Int_{closed}	0	0	257.992

5.2 Results

As can be seen in Table 1, the external vocabulary reduces the out-of-vocabulary rate from 6.15% to 2.80% for the largest of the tested language models and the effect on the recognition accuracies is shown in Fig. 4. One can see that the performance increases with the size of the external vocabulary, but seems to converge at 85.22% achieved with Ext_{300}. The large increase between Int_{closed} and Ext_{20} shows that even a small set of external words can help the recognition substantially. The differences between the four recognitions using Int_{open} (82.73%), Ext_{20} (84.28%), Ext_{300} (85.22%), and Int_{closed} (89.75%) are all statistically significant according to a Student's T-test with $\alpha = 0.05$.

The more words are added, the lower is the out-of-vocabulary rate and more words can potentially be recognized. However, with a larger vocabulary the chance for confusing words also increases. To distinguish between theses two effects, a comparison between the word accuracy rate (WAR) and a normalized word accuracy rate (WAR*) is given in Table 2. The normalized word accuracy rate is the fraction of correctly recognized words out of all words that can be recognized with respect to the vocabulary. Hence, we compute the WAR* as WAR/(1-OOVRate). The addition of a small external dictionary of 20k words increases the absolute and normalized accuracy rate. When further words are added, the normalized accuracy rate decreases. For 300k added words, the benefit of a decreased out-of-vocabulary rate and the risk of recognizing wrong words balance each other out. Hence, a further increase for even larger dictionaries seems unlikely.

The decoding with Ext_{300} was also the limit as far as memory resources on the computer system are concerned. The

[3]http://www.speech.sri.com/projects/srilm/

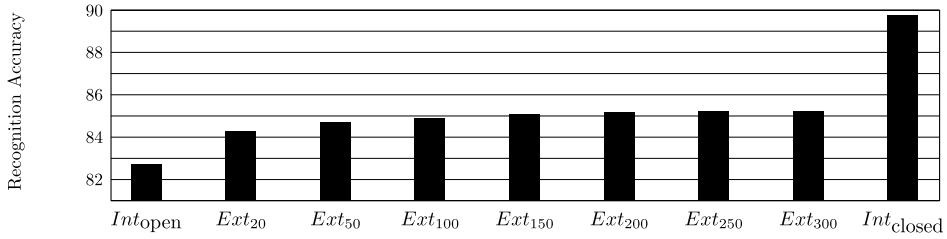

Figure 4: The recognition accuracies using the different language models.

Table 2: The absolute and relative word accuracy rates.

LM	WAR	WAR*
Int_{open}	82.73	88.15
Ext_{20}	84.28	88.44
Ext_{50}	84.68	88.30
Ext_{100}	84.91	88.08
Ext_{150}	85.08	88.00
Ext_{200}	85.17	87.90
Ext_{250}	85.21	87.79
Ext_{300}	85.22	87.67
Int_{closed}	89.75	89.75

experiments were conducted of a cluster of Intel®Xeon® CPU E5-2665 0 with a clock speed of 2.40GHz and 12GB memory. The average time it took to decode a text line was 24.90s for the Int_{open} language model, 27.30s for Ext_{20} and 39.78s for Ext_{300}. That is, although the vocabulary size is increased by factor 15.0, the runtime is only increased by factor 1.5 when comparing Ext_{20} and Ext_{300}.

6. CONCLUSIONS

In this article we described a system for the automatic transcription of historical documents using very large vocabularies gathered from two external sources, the Google N-gram project and an edition of a large, manually transcribed book of the same epoch.

Experiments were conducted on the Old Spanish RODRIGO database. With the inclusion of external language sources, we could significantly reduce the out-of-vocabulary rate from 6.15% to 2.80% (−3.35%) and by doing so increase the recognition rate from 82.73% to 85.22% (+2.49%). The positive influence of a larger vocabulary could be observed up to size of 300k external words. By relying on an efficient token passing algorithm for BLSTM NN recognition, the runtime was only increased by factor 1.5 when using 300k instead of 20k vocabulary words.

This work shows therefore how the drawback of limited available language data for historical documents can be effectively reduced by a massive vocabulary extension. Future work involves experiments with even larger vocabularies and more sophisticated language models, to further increase the final recognition rate.

7. ACKNOWLEDGMENTS

This work has been supported by the European project FP7-PEOPLE-2008-IAPP: 230653 the European Research Council's Advanced Grant ERC-2010-AdG 20100407, the Spanish R&D projects TIN2009-14633-C03-03, RYC-2009-05031, TIN2011-24631, TIN2012-37475-C02-02, MITTRAL (TIN2009-14633-C03-01), Active2Trans (TIN2012-31723) as well as the Swiss National Science Foundation fellowship project PBBEP2_141453.

8. REFERENCES

[1] *El Ingenioso Hidalgo Don Qvixote de la Mancha*, volume 1. `http://www.cervantesvirtual.com/`, Madrid: Gráficas Reunidas, 1928-1931 edition, 1615.

[2] A. Antonacopoulos and A. C. Downton. Special Issue on the Analysis of Historical Documents. *Int'l Journal of Document Analysis and Recognition (IJDAR)*, 9(2-7):75–77, 2007.

[3] B. Barett, M. S. Brown, R. Manmatha, and J. Gehring, editors. *Int'l Conf. on Historic Imaging and Processing*, New York, NY, USA, 2011. ACM Digital Libary.

[4] A. Fischer. *Handwriting Recognition in Historical Documents*. PhD thesis, University of Bern, Switzerland, pages 96–100, 2012.

[5] A. Fischer, E. Indermühle, H. Bunke, G. Viehhauser, and M. Stolz. Ground truth creation for handwriting recognition in historical documents. In *Proc. 9th Int. Workshop on Document Analysis Systems*, pages 3–10, 2010.

[6] A. Fischer, M. Wüthrich, M. Liwicki, V. Frinken, H. Bunke, G. Viehauser, and M. Stolz. Automatic Transcription of Handwritten Medieval Documents. In *Virtual Systems and Multimedia*, pages 137–142, 2009.

[7] V. Frinken, A. Fischer, R. Manmatha, and H. Bunke. A Novel Word Spotting Method Based on Recurrent Neural Networks. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 34(2):211–224, 2012.

[8] J. T. Goodman. A Bit of Progress in Language Modeling: Extended Version. Technical Report MSR-TR-2001-72, Microsoft Resarch, 2001.

[9] A. Graves. RNNLIB: A recurrent neural network library for sequence learning problems. `http://sourceforge.net/projects/rnnl/`.

[10] A. Graves, M. Liwicki, S. Fernández, R. Bertolami, H. Bunke, and J. Schmidhuber. A Novel Connectionist System for Unconstrained Handwriting Recognition. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 31(5):855–868, 2009.

[11] A. Graves and J. Schmidhuber. Framewise Phoneme Classification with Bidirectional LSTM Networks. In *Int'l Joint Conf. on Neural Networks*, volume 4, pages 2047–2052, 2005.

[12] R. Kneser and H. Ney. Improved Bbacking-Off for M-Gram Language Modeling. In *Int'l Conf. Acoustic, Speech, and Signal Processing*, volume 1, pages 181–184, 1995.

[13] A. L. Koerich, R. Sabourin, and C. Suen. Large vocabulary off-line handwriting recognition: A survey. *Pattern Analysis and Applications*, 6:97–121, 2003.

[14] M. Kozielski, D. Rybach, S. Hahn, R. Schlüter, and H. Ney. Open Vocabulary Handwriting Recognition Using Combined Word-Level and Character-Level Language Models. In *Int'l Conf. on Acoustics, Speech, and Signal Processing*, page accepted for publication, 2013.

[15] U.-V. Marti and H. Bunke. Using a statistical language model to improve the performance of an HMM-based cursive handwriting recognition system. *Int. Journal of Pattern Recognition and Artificial Intelligence*, 15:65–90, 2001.

[16] U.-V. Marti and H. Bunke. *Hidden Markov models: Applications in Computer Vision*, chapter Using a Statistical Language Model to Improve the Performance of an HMM-Based Cursive Handwriting Recognition System, pages 65–90. World Scientific Publishing Co., Inc., River Edge, NJ, USA, 2002.

[17] J.-B. Michel, Y. K. Shen, A. P. Aiden, A. Veres, M. K. Gray, W. Brockman, The Google Books Team, J. P. Pickett, D. Hoiberg, D. Clancy, P. Norvig, J. Orwant, S. Pinker, M. A. Nowak, and E. L. Aiden. Quantitative analysis of culture using millions of digitized books. *Science*, 331(6014):176–182, 1 2010.

[18] N. Serrano, F. Castro, and A. Juan. The rodrigo database. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta, may 2010. ELRA.

[19] A. H. Toselli, E. Vidal, and F. Casacuberta. *Multimodal Interactive Pattern Recognition and Applications*. Springer-Verlag, 2011.

[20] M. Wüthrich, M. Liwicki, A. Fischer, E. Indermühle, H. Bunke, G. Viehhauser, and M. Stolz. Language Model Integration for the Recognition of Handwritten Medieval Documents. In *10th Int'l Conf. on Document Analysis and Recognition*, pages 211–215, 2009.