Transliteration of Proper Names in Cross-Lingual Information Retrieval

Paola Virga
Johns Hopkins University
3400 North Charles Street
Baltimore, MD 21218, USA
paola@jhu.edu

Sanjeev Khudanpur
Johns Hopkins University
3400 North Charles Street
Baltimore, MD 21218, USA
khudanpur@jhu.edu

Abstract

We address the problem of transliterating English names using Chinese orthography in support of cross-lingual speech and text processing applications. We demonstrate the application of statistical machine translation techniques to "translate" the phonemic representation of an English name, obtained by using an automatic text-to-speech system, to a sequence of initials and finals, commonly used subword units of pronunciation for Chinese. We then use another statistical translation model to map the initial/final sequence to Chinese characters. We also present an evaluation of this module in retrieval of Mandarin spoken documents from the TDT corpus using English text queries.

1 Introduction

Translation of proper names is generally recognized as a significant problem in many multi-lingual text and speech processing applications. Even when hand-crafted translation lexicons used for machine translation (MT) and cross-lingual information retrieval (CLIR) provide significant coverage of the words encountered in the text, a significant portion of the tokens not covered by the lexicon are proper names and domain-specific terminology (cf., e.g., Meng et al (2000)). This lack of translations adversely affects performance. For CLIR applications in particular, proper names and technical terms are especially important, as they carry the most distinctive information in a query as corroborated by their relatively low document frequency. Finally, in interactive IR systems where users provide very short queries (e.g. 2-5 words), their importance grows even further.

Unlike specialized terminology, however, proper names are amenable to a speech-inspired translation approach. One tries, when writing foreign names in ones own language, to preserve the way it sounds. i.e. one uses an orthographic representation which, when "read aloud" by a speaker of ones language sounds as much like it would when spoken by a speaker of the foreign language — a process referred to as transliteration. Therefore, if a mechanism were available to render, say, an English name in its phonemic form, and another mechanism were available to convert this phonemic string into the orthography of, say, Chinese, then one would have a mechanism for transliterating English names using Chinese characters. The first step has been addressed extensively, for other obvious reasons, in the *automatic speech synthesis* literature. This paper describes a statistical approach for the second step.

Several techniques have been proposed in the recent past for name transliteration. Rather than providing a comprehensive survey we highlight a few representative approaches here. Finite state transducers that implement transformation rules for *back*-transliteration from Japanese to English have been described by Knight and Graehl (1997), and extended to Arabic by Glover-Stalls and Knight (1998). In both cases, the goal is to recognize words in Japanese or Arabic text which hap-

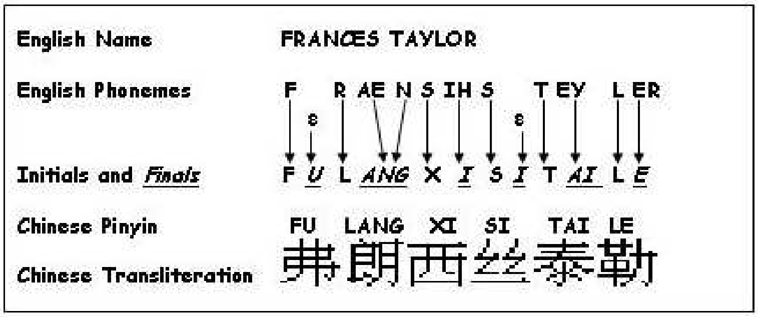

Figure 1: Four steps in English-to-Chinese transliteration of names.

pen to be transliterations of English names. If the orthography of a language is strongly phonetic, as is the case for Korean, then one may use relatively simple hidden Markov models to transform English pronunciations, as shown by Jung et al (2000). The work closest to our application scenario, and the one with which we will be making several direct comparisons, is that of Meng et al (2001). In their work, a set of hand-crafted transformations for *locally* editing the phonemic spelling of an English word to conform to rules of Mandarin syllabification are used to seed a transformation-based learning algorithm. The algorithm examines some data and learns the proper sequence of application of the transformations to convert an English phoneme sequence to a Mandarin syllable sequence. Our paper describes a data driven counterpart to this technique, in which a cascade of two source-channel translation models is used to go from English names to their Chinese transliteration. Thus even the initial requirement of creating candidate transformation rules, which may require knowledge of the phonology of the target language, is eliminated.

We also investigate incorporation of this transliteration system in a cross-lingual spoken document retrieval application, in which English text queries are used to index and retrieve Mandarin audio from the TDT corpus.

2 Translation System Description

We break down the transliteration process into various steps as depicted in Figure 1.

1. Conversion of an English name into a phonemic representation using the Festival[1] speech synthesis system.
2. Translation of the English phoneme sequence into a sequence of generalized initials and finals or GIFs — commonly used sub-syllabic units for expressing pronunciations of Chinese characters.
3. Transformation of the GIF sequence into a sequence of pin-yin symbols without tone.
4. Translation of the pin-yin sequence to a character sequence.

Steps 1. and 3. are deterministic transformations, while Steps 2. and 4. are accomplished using statistical means.

The IBM source-channel model for statistical machine translation (P. Brown et al., 1993) plays a central role in our system. We therefore describe it very briefly here for completeness. In this model, a J-word foreign language sentence $\mathbf{f} = f_1\ f_2\ \ldots\ f_J$ is modeled as the output of a "noisy channel" whose input is its correct I-word English translation $\mathbf{e} = e_1\ e_2\ \ldots\ e_I$, and having observed the channel output $\mathbf{f}$, one seeks *a posteriori* the most likely English sentence

$$\hat{\mathbf{e}} = \arg\max_{\mathbf{e}} P(\mathbf{e}|\mathbf{f}) = \arg\max_{\mathbf{e}} P(\mathbf{f}|\mathbf{e})P(\mathbf{e})$$

The *translation model* $P(\mathbf{f}|\mathbf{e})$ is estimated from a paired corpus of foreign-language sentences and their English translations, and the *language model* $P(\mathbf{e})$ is trained from English text. Software tools

[1] http://www.speech.cs.cmu.edu/festival

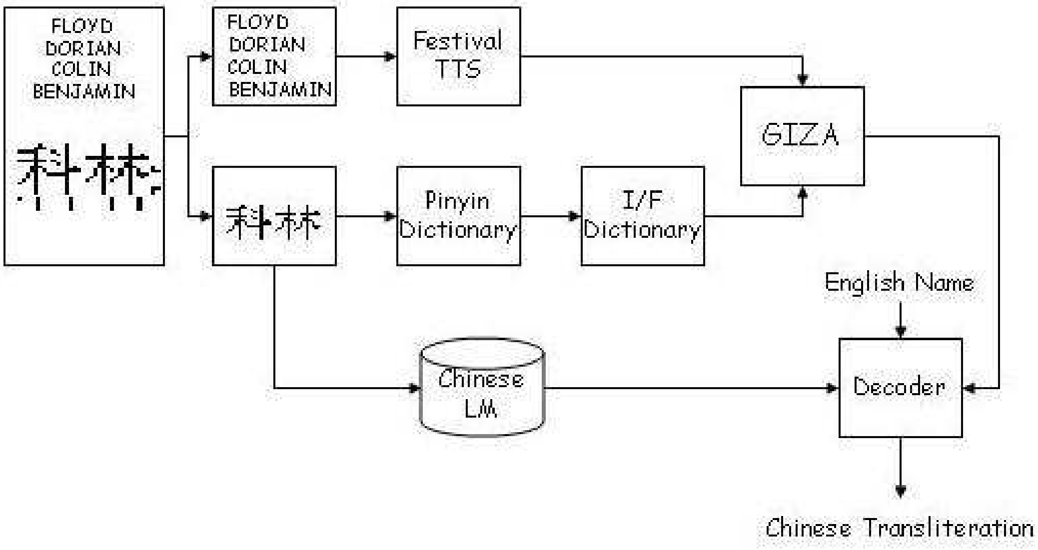

Figure 2: Schematic of a English-to-Chinese name transliteration system.

are available both for training models[2] as well as for *decoding*[3] — the task of determining the most likely translation $\hat{\mathbf{e}}$.

Since we seek Chinese names which are transliteration of a given English name, the notion of words in a sentence in the IBM model above is replaced with phonemes in a word. The roles of English and Chinese are also reversed. Therefore, $\mathbf{f} = f_1\ f_2\ \ldots\ f_J$ represents a sequence of English phonemes, and $\mathbf{e} = e_1\ e_2\ \ldots\ e_I$, for instance, a sequence of GIF symbols in Step 2. described above. The overall architecture of the proposed transliteration system is illustrated in Figure 2.

2.1 Translation Model Training

We have available from Meng et al (2000) a small list of about 3875 English names and their Chinese transliteration. A pin-yin rendering of the Chinese transliteration is also provided. We use the Festival text-to-speech system to obtain a phonemic pronunciation of each English name. We also replace all pin-yin symbols by their pronunciations, which are described using an inventory of generalized initials and finals. The pronunciation table for this purpose is obtained from an elementary Mandarin textbook (Practical Chinese Reader, 1981). The net result is a corpus of 3875 pairs of "sentences" of the kind depicted in the second and third lines of Figure 1. The vocabulary of the English side of this *parallel corpus* is 43 phonemes, and the Chinese side is 58 (21 initials and 37 finals). Note, however, that only 409 of the 21×37 possible initial-final combinations constitute legal pin-yin symbols.

A second corpus of 3875 "sentence" pairs is derived corresponding to the fourth and fifth lines of Figure 1, this time to train a statistical model to translate pin-yin sequences to Chinese characters. The vocabulary of the pin-yin side of this corpus is 282 and that of the character side is about 680. These, of course, are much smaller than the inventory of Chinese pin-yin- and character-sets. We note that certain characters are preferentially used in transliteration over others, and the resulting frequency of character-usage is not the same as unrestricted Chinese text. However, there isn't a distinct set of characters exclusively for transliteration.

For purposes of comparison with the transliteration accuracy reported by Meng et al (2001), we divide this list into 2233 training name-pairs and 1541 test name-pairs. For subsequent CLIR experiments, we create a larger training set of 3625 name-pairs, leaving only 250 names-pairs for intrinsic testing of transliteration performance. The actual training of all translation models proceeds according to a stan-

[2] http://www-i6.informatik.rwth-aachen.de/ och/software/GIZA++.html.

[3] http://www.isi.edu/licensed-sw/rewrite-decoder.

dard recipe recommended in GIZA++, namely 5 iterations of Model 1, followed by 5 of Model 2, 10 HMM-iterations and 10 iterations of Model 4.

2.2 Language Model Training

The GIF language model required for translating English phoneme sequences to GIF sequences is estimated from the training portion of the 3875 Chinese names. A trigram language model on the GIF vocabulary is estimated with the CMU toolkit, using Good-Turing smoothing and Katz back-off. Note that due to the smoothing, this language model does not necessarily assign zero probability to an illegal GIF sequence, e.g., one containing two consecutive initials. This causes the first translation system to sometimes, though very rarely, produce GIF sequences which do not correspond to any pin-yin sequence. We make an ad hoc correction of such sequences when mapping a GIF sequence to pin-yin, which is otherwise trivial for all legal sequences of initials and finals. Specifically, a final `e` or `i` or `a` is tried, in that order, between consecutive initials until a legitimate sequence of pin-yin symbols obtains.

The language model required for translating pin-yin sequences to Chinese characters is relatively straightforward. A character trigram model with Good-Turing discounting and Katz back-off is estimated from the list of transliterated names.

2.3 Decoding Issues

We use the ReWrite decoder provided by ISI, along with the two translation models and their corresponding language models trained, either on 2233 or 3625 name-pairs, as described above, to perform transliteration of English names in the respective test sets with 1541 or 250 name-pairs respectively.

1. An English name is first converted to a phoneme sequence via Festival.
2. The phoneme sequence is translated into an GIF sequence using the first translation model described above.
3. The translation output is corrected if necessary to create a legitimate pin-yin sequence.
4. The pin-yin sequence is translated into a sequence of Chinese characters using a second translation model, also described above.

A small but important manual setting in the ReWrite decoder is a list of *zero fertility words*. In the IBM model described earlier, these are the words e_i which may be "deleted" by the noisy channel when transforming $\mathbf{e}$ into $\mathbf{f}$. For the decoder, these are therefore the words which may be optionally inserted in $\hat{\mathbf{e}}$ even when there is no word in $\mathbf{f}$ of which they are considered a direct translation. For the usual case of Chinese to English translation, these would usually be articles and other function words which may not be prevalent in the foreign language but frequent in English.

For the phoneme-to-GIF translation model, the "words" which need to be inserted in this manner are *syllabic nuclei*! This is because Mandarin does not permit complex consonant clusters in a way that is quite prevalent in English. This linguistic knowledge, however, need not be imparted by hand in the IBM model. One can, indeed, derive such a list from the trained models by simply reading off the list of symbols which have zero fertility with high probability. This list, in our case, is { `-i`, `e`, `u`, `o`, `r`, ü, `ou`, `c`, `iu`, `ie`}.

The second translation system, for converting pin-yin sequences to character sequences, has a one-to-one mapping between symbols and therefore has no words with zero fertility.

2.4 Intrinsic Evaluation of Transliteration

We evaluate the efficacy of our transliteration at two levels. For comparison with the very comparable set-up of Meng et al (2001), we measure the accuracy of the pin-yin output produced by our system after Step 3. in Section 2.3. The results are shown in Table 1, where pin-yin error rate is the edit distance between the "correct" pin-yin representation of the correct transliteration and the pin-yin sequence output by the system.

Translation System	Training Size	Test Size	Pin-yin Errors	Char Errors
Meng et al	2233	1541	52.5%	N/A
Small MT	2233	1541	50.8%	57.4%
Big MT	3625	250	49.1%	57.4%

Table 1: Pin-yin and character error rates in automatic transliteration.

Note that the pin-yin error performance of our fully statistical method is quite competitive with previous results. We further note that increasing the training data results in further reduction of the syllable error rate. We concede that this performance, while comparable to other systems, is not satisfactory and merits further investigation.

We also evaluate the efficacy of our second translation system which maps the pin-yin sequence produced by the previous stages to a sequence of Chinese characters, and obtain character error rates of 12.6%. Thus every correctly recognized pin-yin symbol has a chance of being transformed with some error, resulting in higher character error rate than the pin-yin error rate. Note that while significantly lower error rates have been reported for converting pin-yin to characters in generic Chinese text, ours is a highly specialized subset of transliterated foreign names, where the choice between several characters sharing the same pin-yin symbol is somewhat arbitrary.

3 Spoken Document Retrieval System

Several multi-lingual speech and text applications require some form of name transliteration, cross-lingual spoken document retrieval being a prototypical example. We build upon the experimental infrastructure developed at the 2000 Johns Hopkins Summer Workshop (Meng et al., 2000) where considerable work was done towards indexing and retrieving Mandarin audio to match English text queries. Specifically, we find that in a large number of queries used in those experiments, English proper names are not available in the translation lexicon, and are subsequently ignored during retrieval. We use the technique described above to transliterate all such names into Chinese characters and observe the effect on retrieval performance.

The TDT-2 corpus, which we use for our experiments, contains 2265 audio clips of Mandarin news stories, along with several thousand contemporaneously published Chinese text articles, and English text and audio broadcasts. The articles tend to be several hundred to a few thousand words long, while the audio clips tend to be two minutes or less on average. The purpose of the corpus is to facilitate research in topic detection and tracking and exhaustive relevance judgments are provided for several topics. i.e. for each of at least 17 topics, every English and Chinese article and news clip has been examined by a human assessor and determined to be either on- or off-topic. We randomly select an English article on each of the 17 topics as a query, and wish to retrieve all the Mandarin audio clips on the same topic without retrieving any that are off-topic. For mitigating the variability due to query selection, we choose up to 12 different English articles for each of the 17 topics and average retrieval performance over this selection before reporting any results. We use the query term-selection and translation technique described by Meng et al (2000) to convert the English document to Chinese, the only augmentation being the transliterated names — there are roughly 2000 tokens in the queries which are not translatable, and almost all of them are proper names. We report IR performance with and without the name-transliteration.

We use a different information retrieval system from the one used in the 2000 Workshop (Meng et al., 2000) to perform the retrieval task. A brief description of the system is therefore in order.

3.1 The HAIRCUT System

The Hopkins Automated Information Retriever for Combing Unstructured Text (HAIRCUT) is a research retrieval system developed at the Johns Hopkins University Applied Physics Laboratory. The system was developed to investigate knowledge-light methods for linguistic processing in text retrieval. HAIRCUT uses a statistical language model of retrieval such as the one explored by Hiemstra (2001). The model ranks documents according to the probability that the terms in a query are generated by a document. Various smoothing methods have been proposed to combine the contributions for each term based on the document model and also a generic model of the language. Many have found that a simple mixture model using document term frequencies for the former, and occurrence statistics from a large corpus for the later, works quite well.

McNamee and Mayfield (2001) have shown using HAIRCUT that overlapping character n-grams are effective for retrieval in non-Asian languages (e.g., using n=6) and that translingual retrieval between closely related languages is quite feasible even with-

CLIR System	mean Average Precision		
	No NE Transliteration	Automatic NE Transliteration	LDC NE Look-Up
Meng et al (2001)	0.514	0.522	NA
Haircut	0.501	0.515	0.506

Table 2: Cross-lingual retrieval performance with and without name transliteration

out translation resources of any kind (McNamee and Mayfield, 2002).

For the task of retrieving Mandarin audio from Chinese text queries on the TDT-2 task, the system described by Meng et al (2000) achieved a mean average precision of 0.733 using character bigrams for indexing. On identical queries, HAIRCUT achieved 0.762 using character bigrams. This figure forms the monolingual baseline for our CLIR system.

3.2 Cross-Lingual Retrieval Performance

We first indexed the automatic transcription of the TDT-2 Mandarin audio collection using character bigrams, as done by Meng et al (2000). We performed CLIR using the Chinese translations of the English queries, with and without transliteration of proper names, and compared the standard 11-step mean average precision (mAP) on the TDT-2 audio corpus. Our results and the corresponding results from Meng et al (2001) are reported in Table 2.

Without name transliteration, the performance of the two CLIR systems is nearly identical: a paired t-test shows that the difference in the mAPs of 0.514 and 0.501 is significant only at a p-value of 0.74.

A small improvement in mAP is obtained by the Haircut system with name transliteration over the system without name transliteration: the improvement from 0.501 to 0.515 is statistically significant at a p-value of 0.084. The statistical significance of the improvement from 0.514 to 0.522 by Meng et al (2001) is not known to us. In any event, a need for improvement in transliteration is suggested by this result.

We recently received a large list of nearly 2M Chinese-English named-entity pairs from the LDC. As a pilot experiment, we simply added this list to the translation lexicon of the CLIR system, i.e., we "translated" those names in our English queries which happened to be available in this LDC list. This happens to cover more than 85% of the previously untranslatable names in our queries. For the remaining names, we continued to use our automatic transliterator. To our surprise, the mAP improvement from 0.501 to 0.506 was statistically insignificant (p-value of 0.421) and the reason why the use of the ostensibly correct transliteration most of the time still does not result in any significant gain in CLIR performance continues to elude us.

We conjecture that the fact that the audio has been processed by an automatic speech recognition system, which in all likelihood did not have many of the proper names in question in its vocabulary, may be the cause of this dismal performance. It is plausible, though we cannot find a stronger justification for it, that by using the 10-best transliterations produced by our automatic system, we are adding robustness against ASR errors in the retrieval of proper names.

4 A Large Chinese-English Translation Table of Named Entities

The LDC Chinese-English named entity list was compiled from Xinhua News sources, and consists of nine pairs of lists, one each to cover person-names, place-names, organizations, etc. While there are indeed nearly 2 million name-pairs in this list, a large number of formatting, character encoding and other errors exist in this beta release, making it difficult to use the corpus as is in our statistical MT system. We have tried using from this resource the two lists corresponding to person-names and place-names respectively, and have attempted to augment the training data for our system described previously in Section 2.1. However, we further screened these lists as well in order to eliminate possible errors.

4.1 Extracting Named Entity Transliteration Pairs for Translation Model Training

There are nearly 1 million pairs of person or place-names in the LDC corpus. In order to obtain a *clean* corpus of Named Entity transliterations we performed the following steps:

1. We coverted all name-pairs into a *parallel* corpus of English phonemes on one side and Chinese GIFs on the other by the procedure described earlier.

2. We trained a statistical MT system for translating from English phonemes to Chinese GIFs from this corpus.

3. We then *aligned* all the (nearly 1M) training "sentence" pairs with this translation model, and extracted roughly a third of the sentences with an alignment score above a certain tunable threshold (10^{-6}). This resulted in the extraction of 346860 name-pairs.

4. We divided the set into 343738 pairs for training and 3122 for testing.

5. We estimated a pin-yin language model from the training portion above.

6. We retrained the statistical MT system on this presumably "good" training set and evaluated the pin-yin error rate of the transliteration.

The result of this evaluation is reported in Table 3 against the line "Huge MT (Self)," where we also report the transliteration performance of the so-called Big MT system of Table 1 on this new test set. We note, again with some dismay, that the additional training data did not result in a significant improvement in transliteration performance.

MT System (Data filtered by)	Training Size	Test Size	Pin-yin Errors
Big MT	3625	3122	51.1%
Huge MT (Itself)	343738	3122	51.5%
Huge MT (Big MT)	309019	3122	42.5%

Table 3: Pin-yin error rates for MT systems with varying amounts of training data and different data selection procedures.

We continue to believe that careful data-selection is the key to successful use of this beta-release of the LDC Named Entity corpus. We therefore went back to Step 3 of the procedure outlined above, where we had used alignment scores from an MT system to select "good" sentence-pairs from our training data, and instead of using the MT system trained in Step 2 immediately preceding it, we used the previously built Big MT system of Section 2.1, which we know is trained on a small but clean data-set of 3625 name-pairs. With a similar threshold as above, we again selected roughly 300K name-pairs, being careful to leave out any pair which appears in the 3122 pair test set described above, and reestimated the entire phoneme-to-GIF translation system on this new corpus. We evaluated this system on the 3122 name-pair test set for transliteration performance, and the results are included in Table 3.

Note that significant improvements in transliteration performance result from this alternate method of data selection.

4.2 Cross-Lingual Retrieval Performance — II

We reran the CLIR experiments on the TDT-2 corpus using the somewhat improved entity transliterator described above, with the same query and document collection specifications as the experiments reported in Table 2. The results of this second experiment is reported in Table 4, where the performance of the Big MT transliterator is reproduced for comparison.

Transliterator (Data filtered by)	mean Average Precision	
	No NE	Automatic NE
Big MT	0.501	0.515
Huge MT (Big MT)	—	0.517

Table 4: Cross-lingual retrieval performance with and without name transliteration

Note that the gain in CLIR performance is again only somewhat significant, with the improvement in mAP from 0.501 to 0.517 being significant only at a p-value of 0.080.

5 Concluding Remarks

We have presented a name transliteration procedure based on statistical machine translation techniques and have investigated its use in a cross lingual spoken document retrieval task. We have found small gains in the extrinsic evaluation of our procedure: mAP improvement from 0.501 to 0.517. In a more intrinsic and direct evaluation, we have found ways to gainfully filter a large but noisy training corpus to augment the training data for our models and improve transliteration accuracy considerably beyond our starting point, e.g., to reduce Pin-yin error rates from 51.1% to 42.5%. We expect to further refine the translation models in the future and apply them in other tasks such as text translation.

References

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263-311.

Sung Young Jung, SungLim Hong, and Eunok Paek. 2000. An English to Korean Transliteration Model of Extended Markov Window. *Proceedings of COLING*.

K. Knight and J. Graehl. 1997. Machine Transliteration. *Proceedings of ACL*.

Paul McNamee and Jim Mayfield. 2001. JHU/APL Experiments at CLEF-2001: Translation Resources and Score Normalization. *Proceedings of CLEF*.

Paul McNamee and Jim Mayfield. 2002. Comparing Cross-Language Query Expansion Techniques by Degrading Translation Resources. *Proceedings of SIGIR*.

Helen M. Meng et al. 2000. Mandarin-English Information (MEI): Investigating Translingual Speech Retrieval. Technical Report for the Johns Hopkins Univ. Summer Workshop.

Helen M. Meng, Wai-Kit Lo, Berlin Chen, and Karen Tang. 2001. Generating Phonetic Cognates to Handle Named Entities in English-Chinese Cross-Language Spoken Document Retrieval. *Proceedings of ASRU*.

Practical Chinese Reader, Book I. The Commercial Press LTD. 1981.

Bonnie Glover Stalls and Kevin Knight. 1998. Translating Names and Technical Terms in Arabic Text. *Proceedings of the COLING/ACL Workshop on Computational Approaches to Semitic Languages*.

Djoerd Hiemstra. 2001. Using Language Models for Information Retrieval. Ph.D. thesis,University of Twente, Netherlands.