Statistical Transliteration for English-Arabic Cross Language Information Retrieval

Nasreen AbdulJaleel and Leah S. Larkey
Center for Intelligent Information Retrieval
Computer Science, University of Massachusetts
140 Governors Drive
Amherst, MA 01003-4610
Tel: 1-413-545-3415
Fax: 1-413-545-1789
{nasreen, larkey}@cs.umass.edu

ABSTRACT

Out of vocabulary (*OOV*) words are problematic for cross language information retrieval. One way to deal with OOV words when the two languages have different alphabets, is to *transliterate* the unknown words, that is, to render them in the orthography of the second language. In the present study, we present a simple statistical technique to train an English to Arabic transliteration model from pairs of names. We call this a *selected n-gram* model because a two-stage training procedure first learns which n-gram segments should be added to the unigram inventory for the source language, and then a second stage learns the translation model over this inventory. This technique requires no heuristics or linguistic knowledge of either language. We evaluate the statistically-trained model and a simpler hand-crafted model on a test set of named entities from the Arabic AFP corpus and demonstrate that they perform better than two online translation sources. We also explore the effectiveness of these systems on the TREC 2002 cross language IR task. We find that transliteration either of OOV named entities or of all OOV words is an effective approach for cross language IR.

Categories and Subject Descriptors

H.3.3 **[Information Storage and Retrieval]**: Information Search and Retrieval

General Terms

Algorithms, Performance, Design, Experimentation, Languages

Keywords

Cross language information retrieval, Statistical transliteration, Out of vocabulary words, Named entities

CIKM'03, November 3–8, 2003, New Orleans, Louisiana, USA.

1. INTRODUCTION

Out of vocabulary (OOV) words are a common source of errors in cross language information retrieval (CLIR). Bilingual dictionaries are often limited in their coverage of named-entities, numbers, technical terms and acronyms. There is a need to generate translations for these "on-the-fly" or at query time.

A significant proportion of OOV words are named entities and technical terms. Typical analyses find around 50% of OOV words to be named entities [6][8]. Yet these can be the most important words in the queries. Larkey et al [16] showed that cross language retrieval performance (average precision) reduced more than 50% when named entities in the queries were not translated.

Variability in the English spelling of words of foreign origin may contribute to OOV errors. Whitaker [21], for example, identifies 32 different English spellings for the name of the Libyan leader Muammar Gaddafi.

When the query language and the document language share the same alphabet it may be sufficient to use the OOV word as its own translation. However, when the two languages have different alphabets, the query term must somehow be rendered in the orthography of the other language. The process of converting a word from one orthography into another is called *transliteration*.

Foreign words often occur in Arabic text as transliterations. This is the case for many categories of foreign words, not just proper names but also technical terms such as disease names like *Bilharzia* and common words such as *caviar*, *telephone*, *television*, *computer* and *internet*. Words of foreign origin are often transliterated into English as well. Examples of these are Arabic proper names and words like *hummus* and *hookah*.

There is great variability in the Arabic rendering of foreign words, especially named entities. Although there are spelling conventions, there isn't one "correct" spelling. We have observed multiple spellings for a word even within the same document. Listed below are 6 different spellings for the name *Milosevic* found in one collection of news articles.

Milosevic	ميلوسيفيتش	Mylwsyfytsh
	ميلوسفيتش	Mylwsfytsh

ميلو زفيتش	Mylwzfytsh
ميلو زيفيتش	mylwzyfytsh
ميلسيفيتش	mylsyfytsh
ميلوسيفتش	mylwsyftsh

This variation in spelling implies that a translation source that generates multiple Arabic spellings would be useful for CLIR. Statistical transliteration can be used to generate many alternative spellings and therefore lends itself well to the task of generating translations for CLIR.

The transliteration problem is amenable to standard statistical translation model approaches. Statistical transliteration is a special case of statistical translation, in which the words are individual characters.

In the present study, we present a simple statistical technique for English to Arabic transliteration. This technique requires no heuristics or linguistic knowledge of either language. We evaluate the transliteration system on a test set of proper names from the Arabic AFP corpus used in the CLIR track for TREC 2001 and 2002 [11]. We also compare several available sources of named-entity translations from English to Arabic with the output of our transliterator and demonstrate that our system performs better. We explore the effectiveness of hand-crafted and automatically-trained transliteration models on the TREC2002 CLIR task. We compare three approaches to using transliterations in CLIR: adding transliterations of all named entities, adding transliterations of OOV named entities, and adding transliterations of all OOV query words.

2. PREVIOUS WORK

Although organizations that provide online translation from English to Arabic appear to include English/Arabic transliteration systems, little is published about them. [1][2]. No information is available about how they generate transliterations, or how well they work. Darwish et al. [9] described a transliterator used for TREC-2001, but provided no evaluation of it effectiveness.

Most prior work in Arabic-related transliteration has been for the purpose of machine translation, and for Arabic/English transliteration. Arbabi et al. [5] developed a hybrid neural network and knowledge-based system to generate multiple English spellings for Arabic person names. Knight and Graehl [13] developed a five stage statistical model to do *back transliteration*, that is, recover the original English name from its transliteration into Japanese Katakana. Stalls and Knight [20] adapted this approach for back transliteration from Arabic to English of English names. These systems are very complex, involving a great deal of human design, probably because they were dealing with a more difficult problem than that of forward transliteration. However, as the Milosevic example reveals, forward transliteration for information retrieval is not as simple as the problem of forward transliteration for machine translation, in which one reasonable transliteration is good enough. Al-Onaizan and Knight [3] have produced a simpler Arabic/English transliterator and evaluates how well their system can match a source spelling. Their work includes an evaluation of the transliterations in terms of their reasonableness according to human judges. None of these studies measures their performance on a retrieval task or on other NLP tasks.

Fujii and Ishikawa [10] describe a transliteration system for English-Japanese cross language IR that requires some linguistic knowledge. They evaluate the effectiveness of their system on an English-Japanese cross language IR task. The problem they look at is somewhat different from the one we address, as they only attempted to produce one acceptable translation per word. In Japanese, foreign words are written in a special orthography so it is easy to select which words to transliterate. Our research differs in that our system requires no linguistic knowledge or heuristics. We perform a similar evaluation of our system on the retrieval task and also compare our system with other bilingual resources.

3. TRANSLITERATION METHODS

The selected n-gram transliteration model is a generative statistical model that produces a string of Arabic characters from a string of English characters. The model is a set of conditional probability distributions over Arabic characters and NULL, conditioned on English unigrams and selected n-grams. Each English character n-gram e_i can be mapped to an Arabic character or sequence a_i with probability $P(a_i|e_i)$. In practice, most of the probabilities are zero. For example, the probability distribution for the English character *s* might be: P(س|*s*) = .61, P(ز|*s*) = .19, P(ص|*s*) =.10.

In addition to the individual letters of the alphabet (unigrams), the source language (English) symbol inventory includes begin and end symbols and some n-grams, for example, *sh*, *bb*, and *eE* (*e* at the end of a word).

The model is trained from lists of proper name pairs in English and Arabic, via two alignment stages, the first of which is used to select n-grams for the model, and the second which determines the translation probabilities for the n-grams. For the alignments, we used GIZA++ [12], an extension of the program GIZA, which is part of the Statistical Machine Translation toolkit EGYPT. Its operation is described by Och and Ney [17].

GIZA++ was designed for word alignment of sentence-aligned parallel corpora. We used it to do character-level alignment of word pairs, treating n-grams as words. GIZA++ performed well on this task. A small sample of alignments was examined manually. All of the word-pairs in this sample were correctly aligned. It was able to align with NULL such silent English characters as final *e*, and short vowels that are not always rendered in Arabic. It was also able to align an English symbol with a sequence of Arabic characters when necessary (as in *x* → كس). We used GIZA++ only for alignment, not for building the model.

For training, we started with a list of 125,000 English proper nouns and their Arabic translations from NMSU [4]. English words and their translations were retained only if the English word occurred in a corpus of AP News Articles from 1994-1998. There were 37,000 such names. Arabic translations of these 37,000 names were also obtained from the online bilingual translation systems, Almisbar [2] and Tarjim [1]. The training sets thus obtained are called nmsu37k, almisbar37k

and tarjim37k. We also trained on a combination of all three lists.

Note that the training sets were not cleaned up to contain only transliterations, but in some cases, the lowest scoring alignments were dropped from the training set.

The models were built by executing the following sequence of steps on each training set:

1. The training list was normalized. English words were normalized to lower case, and Arabic words were normalized by removing diacritics, replacing أ, إ and آ with bare alif ا, replacing final ى with ي, and replacing final ة with ه. The first character of a word was prefixed with a Begin symbol, *B,* and the last character was suffixed with an End symbol, *E*.
2. The training words were segmented into unigrams and the Arabic-English word pairs were aligned using GIZA++, with Arabic as the source language and English as the target language.
3. The instances in which GIZA++ aligned a sequence of English characters to a single Arabic character were counted. (The reverse, many Arabic characters mapped to a single English character, hardly ever occurred.). The 50 most frequent of these character sequences, or n-grams, were added to the English symbol inventory.
4. The English training words were resegmented based on the new symbol inventory, that is, if a character was part of an n-gram, it was grouped with the other characters in the n-gram. If not, it was rendered separately. For example:

   Ashworth → A sh w o r th
5. GIZA++ was used to align the above English and Arabic training word-pairs, with English as the source language and Arabic as the target language.
6. The transliteration model was built by counting up alignments from the GIZA++ output and converting the counts to conditional probabilities. Alignments below a probability threshold of 0.01 were removed and the probabilities were renormalized.

To generate Arabic transliterations for an English word, w_e, the word is first segmented according to the n-gram inventory. For each segment, all possible transliterations, w_a, are generated. Each word transliteration receives a score as follows which allows the transliterations to be ranked:

$$P(w_a \mid w_e, w_a \in A) = P(w_a \mid w_e) * P(w_a \in A)$$

where $$P(w_a \mid w_e) = \prod_i P(a_i \mid e_i)$$

and $$P(w_a \in A) = \prod_i P(a_i \mid a_{i-1})$$

$P(w_a \in A)$ is the probability of the word, w_a conforming to the spelling patterns in our sample of Arabic names. It is computed using a letter bigram model of general Arabic as the product of the probabilities of each letter bigram in w_a.

A hand-crafted model was developed to provide a high-quality benchmark against which to measure the performance of the automatically trained transliteration models. This model included most commonly occurring English character n-grams that function as units, such as *ll*, *ch*, *sh*, *tio* etc. The handcrafted model can be seen in the Appendix. The automatically trained model can be seen at [6].

4. EXPERIMENTS

The output of the transliteration models were evaluated in two different ways. The first evaluation uses a measure of *translation accuracy* described below, which measures the correctness of transliterations generated by the models, using the spellings found in the AFP corpus as the standard for correct spelling. The second kind of evaluation uses a cross language information retrieval task and looks at how retrieval performance changes as a result of including transliterations in query translations.

4.1 Translation Accuracy

This evaluation was performed on a test list of 815 words. The test list was built by selecting documents at random from the AFP Arabic corpus, and taking all the named entities from those documents that could be translated by transliteration and which did not occur in the training set. English translations were obtained for these by hand.

There is more than one Arabic translation in the test set for some English test words. However, the test list is not exhaustive in its coverage of translations for the test words. There may be more translations for these words in the corpus that are not included in the test set.

For each test word, all possible transliterations were generated and ranked by the model. Only words that occurred in the reference AFP corpus were retained. From this ranked list we computed a *translation accuracy* measure relative to a ranked list of the *n* highest scoring transliterations for each English word. The translation accuracy is the number of test words for which a correct translation appeared within the top *n* transliterations. This is appropriate for an IR task, because it is often sufficient to simply get one correct translation for a word. Translation accuracy differences were tested for significance using a two-tailed sign test with a cutoff of 0.05.

In what follows we present the results of transliterating the test words using the selected n-gram model, trained using a large number of training pairs from a mixture of sources. We explore the effect of the training corpus by comparing these results with models obtained by training on single sources. We also compare the selected n-gram model with a hand-crafted model. We compare the performance of these two models with translations obtained from various online sources. We look at the effect of using n-grams vs. unigrams. Finally, we investigate how large a training set is required to train this model.

4.1.1 Comparison of training data sources

We have found that different English Arabic text sources and resources follow different spelling conventions, so that training on different sources results in models that generate different transliterations. We started with the assumption that it was best to use a variety of sources for training, so that more spelling variations might be covered. Table 1 shows

translation accuracy for ranked lists of 1, 5, and 20 transliterations, using models derived from different sources of training data. For comparison, the accuracy of the hand-crafted model described in Section 3 is included.

Table 1: Translation accuracy of selected n-gram model automatically trained on different sources of training data

Training Source	Top 1	Top 5	Top 20
Mixture	68.9%	84.8%	89.8%
Almisbar37k	69.3%	85.5%	88.3%
Tarjim37k	63.8%	78.9%	82.2%
Nmsu37k	65.6%	81.4%	82.6%
Hand-crafted	71.2%	89.8%	93.6%

The models perform very well. For top 1, the hand crafted model is significantly better than automatic models trained on Tarjim or NMSU data, but not significantly different from the other two models. For top 5, the hand crafted model is better than all the other models. Of the automatically trained models, those trained on the mixture of sources, or on Almisbar alone, are significantly better than those trained on Tarjim or on NMSU translations. For simplicity, we use the model trained on Almisbar data for the rest of the work reported here.

4.1.2 Comparison of different translation sources

Given that online translation sources like Almisbar and Tarjim exist and are available, why not use them directly, rather than training models from them? First, they are difficult to access within a CLIR system. Second, they provide only one translation for each word, and we have found it advantageous to use multiple translations. Finally, our model gives better translations, as can be seen in Table 2.

Table 2: Translation accuracy of selected n-gram model compared with translations from available translation sources

Translation Source	Top 1	Top 5	Top 20
Selected n-gram model	69.3%	85.5%	88.3%
Almisbar	55.0%	*multiple translations not available for online sources*	
Tarjim	50.7%		
NMSU	29.1%		

The online translation engines, Almisbar and Tarjim, were queried for translations of the 815 English test words. NMSU's translations were obtained for those words that existed on the list. Each of these sources provided only one translation for most words. The selected n-gram model performed significantly better than any of the online sources, even when considering only the top-scoring transliteration.

It is informative to look at an example that illustrates the spelling variations found in these sources. The rows of Table 3 list 6 distinct Arabic spellings of the name *Clinton*. Each column indicates a source of translations. An *x* in a cell means that the indicated source used that spelling.

Table 3: Arabic spellings of *Clinton* from different sources

Arabic Spelling	Pronun-ciation Guide	AFP	NMSU	Tarjim	Almisbar	UN	Translit 1	Translit 5
كلينتون	klyntwn	x		x		x	x	x
كلينتن	klyntn	x*						x
كلينطون	klynTwn		x					x
كلنتن	klntn				x			
كلنتون	klntwn					x		x
كلاينتون	klAyntwn							x

* This spelling was rare, found in only 6 AFP documents

Five of the six spellings (all except the last) are reasonable, and consistent with the way many other English names are rendered in Arabic. However, only the first is useful for retrieval from the AFP collection. This example is particularly striking because one might expect the spelling of such a widely used name to be fairly standardized.

4.1.3 Selected n-grams vs. unigrams

In this section we evaluate the contribution of the selected n-gram segments by comparing the selected n-gram model with a unigram model. The training of the unigram model was carried out like that of the selected n-gram model, except that the first stage alignment (whose purpose was to find n-grams) was skipped. The second stage alignment was carried out using unigrams as segments.

Table 4: Accuracy of selected n-gram vs. unigram models

Model	Top 1	Top 5	Top 20
Selected n-gram	69.3%	85.5%	88.3%
Unigram	53.9%	75.0%	83.8%

The results in Table 4 illustrate the importance of including context provided by n-grams for English-Arabic transliteration. As expected, performance degrades when n-grams are not used in the model.

4.1.4 Training Set Sizes

The models above were trained with 37,000 word pairs, far more than is probably necessary. In this section we investigate training set sizes to see how large a training set is required. Training examples were randomly selected to fill sets of sizes 50, 100, 200, 500, 1000, 2000, 3000, 4000, and 5,000. Since the curve was so obviously flat, we did not include larger training sets.

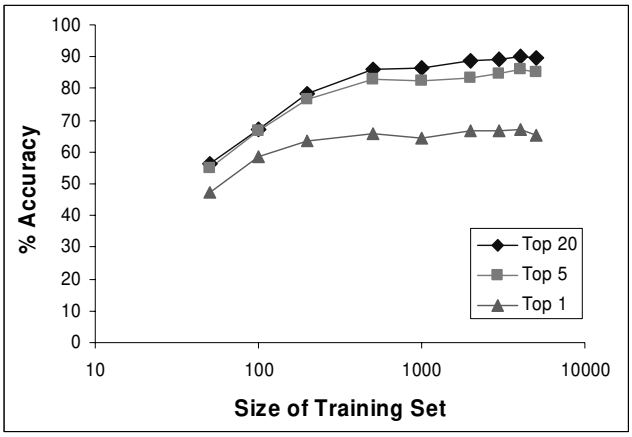

Figure 1: Translation accuracy on training sets of different sizes

Figure 1 shows the translation accuracy obtained by models trained with this range of training set sizes. Translation accuracy was found to increase with training set size for models trained on data sets of 50 to 500 words, and then levels off after that point. The performance of the model trained on 500 words is similar to that of the model trained on the full training set of 37,000 words. This indicates that good quality transliteration models can be trained from small data sets using the selected n-gram model approach.

4.2 IR Evaluation

In this section we assess how well the translations generated by the transliteration models work in the context of information retrieval, the task for which the models were developed. The work in the previous sections showed that when large numbers of transliterations were generated, the sets usually include the correct translations. There is a danger, however, that the large number of other, incorrect translations included in a query can have a deleterious effect on retrieval performance. In order to address this issue, we carried out retrieval experiments using translations from a baseline dictionary, described below, and experimented with several ways of including transliterations into the query translations:

1. Transliterate all the named entities in the query, whether or not they were found in the dictionary.
2. Transliterate only named entities for which translations were not found in the dictionary.
3. Transliterate any words for which translations were not found in the dictionary.

These experiments compare two sources of transliteration – the hand crafted transliteration model, and the automatically trained selected n-gram model evaluated above. In all cases, the top 20 transliterations were included. However, on average, only around 5 of these occurred in the corpus, so the others would have no effect on retrieval.

4.2.1 Baseline Dictionary

The baseline dictionary is an English-Arabic bilingual lexicon used for our TREC2001 work [15]. Most of the words in this baseline dictionary were obtained by querying an online bilingual dictionary [19] using a cgi script that requested an Arabic to English dictionary entry for each Arabic word in the AFP corpus. Each of these entries contained many translations for each word. This online dictionary had very few entries for named entities. We supplemented it with a small bilingual lexicon derived from an English list of world cities and countries found on the web [22]. The list included the names of most countries, their capitals, and a few other major cities. To get the Arabic translations, we used an online machine translation engine, *Sakhr* SET, an older version of *Tarjim* [1] that was available at that time. This list of place names (and only this list, which was made independently of the queries) was hand corrected by an Arabic speaking consultant. Aside from these place names, the dictionary had very few names, but was otherwise adequate. Used in conjunction with good normalization and stemming, this dictionary allowed us to perform well on the TREC 2001 evaluation [15], in which queries contained few named entities and no person names.

4.2.2 Cross Language IR Method

All experiments were carried out using the TREC collection of 383,872 Arabic newspaper articles from the Agence France Presse (AFP), and the 50 TREC 2002 topics, 26-75. Unlike TREC 2001, these queries contain many place and person names, and therefore provide an appropriate test for the transliteration of OOV words. Queries were formed from the title and description fields of the English version of the topics.

For cross language querying of the Arabic collection, we used structural query translation [6], sometimes called the Pirkola method [18], a dictionary-based query translation method in which multiple translations of a term are treated as synonyms. This has the effect of treating the set of translations as a single term in retrieval, whose term frequency is the sum of frequencies of all the different translations, and whose document frequency is the number of documents in the union of the sets of documents containing each translation.

A detailed description of how we performed cross language retrieval experiments with structural query translation can be found in [14]. Here, we include only an overview, with details relevant to the present research.

Basic retrieval experiments contained the following steps:

- English queries were tokenized, lower cased, and stop words were removed.
- Query translation: Each English word was looked up in the baseline dictionary. All the alternative translations for each word were added to the query as synonyms.
- The translated Arabic query was submitted to the AFP collection.
- Standard recall/precision measures were calculated on the ranked list of 1000 retrieved documents. We report uninterpolated average precision, and precision at 10, 20, and 30 documents.

4.2.3 Results of IR Experiments

In this section we investigate retrieval performance when translations for query words are obtained via transliteration. The results for the hand crafted model can be seen in Table 5 and the results for the automatically trained model can be seen in Table 6. The tables show average precision and precision at 10, 20, and 30 documents, using the baseline dictionary, and adding transliterations for query words under three different strategies. The heading *+names* indicates that transliterations were added for all named entities in the queries, whether or not they were found in the dictionary. The heading *+OOV names* indicates that transliterations were added for those named entities in the queries which had no translation in the dictionary. The heading *+OOV words* indicates that transliterations were added for any query words which had no translation in the dictionary, regardless of whether they were named entities or not. The numbers in italics indicate the percent change relative to the baseline in the same row. If the percent change is grayed out, the difference is not statistically significant. If the percent change is shown in black, then it is significant according to the Wilcoxon signed ranks test with $p < .05$.

For the hand-crafted model, adding transliterations produces a significant increase in precision at 10, 20, and 30 docs under any of the three strategies. Average precision shows a significant increase under the two strategies that add transliterations only for items that do not have entries in the dictionary, that is, adding OOV names or adding OOV words. The results for the automatically-trained model show more clearly that it is better to add transliterations only for words or names which do not already have translations in the dictionary.

Table 5: Precision in 50 unexpanded TREC2002 queries, using transliterations from the hand-crafted model.

	Precision and Percent change over baseline			
Precision	**Base line**	**+names**	**+OOV names**	**+OOV words**
Average	.1494	.2276 *(+52.4)*	.2341 *(+56.8)*	.2434 *(+62.9)*
at 10 docs	.2300	.3200 *(+39.1)*	.3340 *(+45.2)*	.3460 *(+50.4)*
at 20 docs	.2010	.2960 *(+47.3)*	.2980 *(+48.3)*	.3060 *(+52.2)*
at 30 docs	.1980	.2740 *(+38.4)*	.2780 *(+40.4)*	.2820 *(+42.4)*

Table 6: Precision in 50 unexpanded TREC2002 queries, using transliterations from the selected n-gram model.

	Precision and Percent change over baseline			
Precision	**Base line**	**+names**	**+OOV names**	**+OOV words**
Average	.1494	.1914 *(+28.1)*	.2034 *(+36.2)*	.2129 *(+42.5)*
at 10 docs	.2300	.2780 *(+20.9)*	.3040 *(+32.2)*	.3140 *(+36.5)*
at 20 docs	.2010	.2550 *(+26.9)*	.2660 *(+32.3)*	.2780 *(+38.3)*
at 30 docs	.1980	.2427 *(+22.6)*	.2493 *(+25.9)*	.2553 *(+29.0)*

Not shown in the table, a direct comparison between the automatically-trained model and the hand-crafted model shows that the handcrafted model yields significantly higher average precision than the automatically-trained model.

Readers who are familiar with the TREC CLIR evaluation may notice that our average precision values are low. This is due to two factors. The first factor is the dictionary. NIST provided a dictionary derived from a huge parallel UN corpus, which covered most query terms, including large numbers of names. In order to distinguish our three strategies we required a dictionary with less complete coverage. As mentioned above, this is not an artificially impoverished dictionary. Rather it is typical of a dictionary not derived from a parallel corpus.

The second factor is query expansion. The results above did not include query expansion, which improves retrieval performance greatly on these queries. We repeated these experiments with query expansion, using a collection of news articles from 1994 through 1998 in the Linguistic Data consortium's NA News corpus to expand the English queries, and the AFP corpus to expand the Arabic queries.

The results did indeed show higher overall levels of precision when queries were expanded. However, the pattern of results did not change. Adding transliterations for OOV names or words improved performance. Adding transliterations for all names did not significantly improve performance.

5. DISCUSSION

The automatically trained models sometimes generated transliterations with common spelling errors like missing letters. These actually occur in the corpus but would not be found in a dictionary and would not be generated by the hand-crafted model. A few examples are shown below, with the missing letters in parenthesis:

Afghanistan	افـانـستـان	af(g)anstan
Clinton	كـلـيـنـون	klin(t)wn
Atlanta	تـلانـتـا	(a)tlanta

These translations would occasionally allow the retrieval of documents that would be missed by more "accurate" methods. One source of retrieval errors is false hits, when transliterations match the wrong word. False hits were more common for short words than for long words, thus in the future one might want to generate more alternatives for long words, or avoid transliterations for very short (e.g. 3 letter) words.

Another possible direction for the future would be to train separate models for words of Arabic origin and words of other origin.

One could also train a back-transliteration model based on an approach similar to the selected n-gram model.

For Arabic, the hand-crafted model performed better than the automatically-trained model, and could be further improved via some error analysis and automatic tuning of weights. However, the automatic selected n-gram approach would be useful when dealing with a new, unfamiliar language.

6. CONCLUSIONS

We have demonstrated a simple technique for statistical transliteration that works well for cross-language IR, in terms

of accuracy and retrieval effectiveness. The results of our experiments support the following generalizations:

- Good quality transliteration models can be generated automatically from reasonably small data sets.
- A hand-crafted model performs slightly better than the automatically-trained model
- The quality of the source of training data affects the accuracy of the model.
- Context dependency is important for the transliteration of English words. The selected n-gram model is more accurate than the unigram model.
- Results of the IR evaluation confirm that transliteration can improve cross-language IR. Further, it is a reasonable strategy to transliterate out-of-vocabulary named entities, or to transliterate out-of-vocabulary words, without requiring any knowledge of which words are named entities. These results do not suggest either alternative as a better choice. However, it is not a good strategy to transliterate names that are already translated in the dictionary.

7. ACKNOWLEDGMENTS

This work was supported in part by the Center for Intelligent Information Retrieval and in part by SPAWARSYSCEN-SD grant numbers N66001-99-1-8912 and N66001-02-1-8903. Any opinions, findings and conclusions or recommendations expressed in this material are the author(s) and do not necessarily reflect those of the sponsor.

8. REFERENCES

[1] Ajeeb online translation engine. http://tarjim.ajeeb.com/ajeeb/

[2] Al Misbar. http://www.almisbar.com/salam_trans.html

[3] Al-Onaizan, Y. and Knight, K. Machine translation of names in Arabic text. Proceedings of the ACL conference workshop on computational approaches to Semitic languages, 2002.

[4] Arabic Proper Names Dictionary from NMSU. http://crl.nmsu.edu/~ahmed/downloads.html[1]

[5] Arbabi, Mansur, Scott M. Fischthal, Vincent C. Cheng, and Elizabeth Bar. 1994. Algorithms for Arabic name transliteration. IBM Journal of research and Development, 38(2):183-193.

[6] Automatically-trained Transliteration Model. http://www.cs.umass.edu/~nasreen/automatic_model.txt

[7] Ballesteros, L. and Croft, W.B. Resolving ambiguity for cross-language retrieval. SIGIR '98, 64-71, 1998.

[8] Davis, M.W. and Ogden, W.C. Free resources and advanced alignment for cross-language text retrieval. In *Proceedings of the sixth text retrieval conference (TREC-6)*, E. M. Voorhees and D. K. Harman (eds.). Gaithersburg: NIST Special Publication 500-240, 385-394, 1998.

[9] Darwish, Kareem, David Doermann, Ryan Jones, Douglas Oard and Mika Rautiainen. 2001. TREC-10 experiments at Maryland: CLIR and video. In TREC 2001. Gaithersburg: NIST. http://trec.nist.gov/pubs/trec10/t10_proceedings.html

[10] Fujii, Atsushi and Tetsuya, Ishikawa. Japanese/English Cross-Language Information Retrieval: Exploration of Query Translation and Transliteration. Computers and the Humanities, Vol.35, No.4, pp.389-420, 2001

[11] Gey, F. C. and Oard, D. W. 2001. The TREC-2001 cross-language information retrieval track: Searching Arabic using English, French, or Arabic queries. In TREC 2001. Gaithersburg: NIST. http://trec.nist.gov/pubs/trec10/t10_proceedings.html

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

[13] Knight, Kevin and Graehl, Jonathan. 1997. Machine transliteration. In Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics, pp. 128-135. Morgan Kaufmann.

[14] Larkey, L. S., Allan, J., Connell, M. E., Bolivar, A., & Wade, C. UMass at TREC 2002: Cross language and novelty tracks, to appear in *The Eleventh Text REtrieval Conference (TREC 2002)*. Gaithersburg: NIST, 2003.

[15] Larkey, L. S., & Connell, M. E. Arabic Information Retrieval at UMass in TREC-10, *The Tenth Text Retrieval Conference, TREC 2001*. Gaithersburg: NIST, 562-570, 2002.

[16] Larkey, Leah, Nasreen AbdulJaleel, and Margaret Connell. 2003. What's in a Name?: Proper Names in Arabic Cross Language Information Retrieval, CIIR Technical Report, IR-278 .

[17] Och, Franz Josef and Hermann Ney. October 2000. Improved Statistical Alignment Models. Proc. of the 38th Annual Meeting of the Association for Computational Linguistics, pp. 440-447, Hong Kong, China.

[18] Pirkola, A. The effects of query structure and dictionary setups in dictionary-based cross-language information retrieval. In proceedings of SIGIR 98 (Melbourne, Australia, Aug 1998), ACM Press 55-63.

[19] Sakhr multilingual dictionary at http://dictionary.ajeeb.com/en.htm

[20] Stalls, Bonnie Glover and Kevin Knight. 1998. Translating names and technical terms in Arabic text. http://citeseer.nj.nec.com/glover98translating.html

[21] Whitaker, B. Arabic words and the Roman alphabet. http://www.al-bab.com/arab/language/roman1.ht

[1] This list used to be available on the web, but has subsequently been removed

[22] World cities.
http://www.fourmilab.ch/earthview/cities.html

9. APPENDIX

Hand-crafted Transliteration Model. Each line shows an English segment, and the Arabic segments which can replace it, along with their probabilities. *B* is a begin marker, *E* is an end marker. To save space, some segments with the same transliteration probability distribution (e.g. b and bb) are shown on the same line. A transliteration of *0* represents a null transliteration.

Ba	ا	0.9	ع	0.1				
Bau	ا	0.8	او	0.2				
Be	ا	0.35	0	0.1	ي	0.25	اي	0.3
Bi	ا	0.7	اي	0.2	ع	0.1		
Bmc	مـاك	0.9	مك	0.1				
Bo	ا	0.3	او	0.7				
Bu	ا	0.8	او	0.2				
Bwr	ر	1.0						
A	ا	0.6	0	0.2	ي	0.1	ع	0.1
aE	ه	0.6	ا	0.4				
ai	ي	0.5	اي	0.5				
alk	وك	0.9	الـك	0.1				
au	ا	0.4	او	0.4	و	0.2		
B,bb	ب	1.0						
C	ك	0.9	تـش	0.1				
cc	ك	1.0						
ce	س	0.8	سي	0.2				
ci,cy	س	0.2	سي	0.8				
ch	تـش	0.8	ش	0.2				
ck	ك	1.0						
D	د	0.99	0	0.01				
dd	د	1.0						
E	0	0.6	ا	0.1	ي	0.3		
eE	0	0.9	ه	0.1				
ea	ي	0.9	يـا	0.1				
ee	ي	1.0						
ey	اي	0.8	ي	0.2				
F,ff,ph	ف	1.0						
G	غ	0.5	ج	0.4	ق	0.1		
ge	ج	0.8	غ	0.2				
gi	جي	0.8	غي	0.2				

gg	غ	0.8	ج	0.2				
gh	0	0.3	ف	0.35	غ	0.35		
gn	نـي	0.3	غن	0.3	جن	0.2	ن	0.2
h	ه	0.8	0	0.1	ح	0.1		
i	ي	0.6	0	0.2	اي	0.2		
ie	ي	0.7	اي	0.3				
j	ج	0.9	ي	0.1				
k,kk	ك	1.0						
kh	خ	1.0						
l	ل	1.0						
ll	ل	0.8	لـل	0.1	ي	0.1		
m,mm	م	1.0						
n,nn	ن	1.0						
o	و	0.7	ا	0.1	0	0.2		
ois	وا	0.8	وس	0.1	ويـس	0.1		
oo	و	1.0						
ou	و	0.6	او	0.4				
ough	او	0.4	وف	0.2	و	0.4		
oughE	ه	0.8	وف	0.2				
p,pp	ب	1.0						
q	ك	0.5	ق	0.5				
qu	كـو	0.6	ك	0.3	ق	0.1		
r,rr	ر	1.0						
s	س	0.6	ز	0.2	ص	0.2		
sch	ش	0.8	ستش	0.2				
sh	ش	1.0						
ss	س	0.8	ص	0.2				
sE	س	0.6	ز	0.4				
t	ت	0.7	ط	0.3				
th	ت	0.3	ث	0.4	ذ	0.3		
tio	ش	0.8	شيـو	0.2				
tt	ت	0.9	ط	0.1				
u	و	0.8	0	0.2				
ueE	0	0.8	و	0.2				
v	ف	0.8	و	0.2				
w	و	0.9	ف	0.1				
wr	ر	0.8	ور	0.2				
x	كس	0.9	خ	0.1				
y	ي	0.8	و	0.1	اي	0.1		
z	ز	0.8	س	0.2				