Transliteration using phrase based SMT approach on substrings

Sara Noeman

IBM
Cairo TDC
Cairo

Abstract

Translation of named entities (NEs), such as person names, organization names and location names is crucial for cross lingual information retrieval, machine translation, and many other natural language processing applications. Newly named entities are introduced on daily basis in newswire and this greatly complicates the translation task.

Named Entities translation between languages having different orthographic basis is more complex than translation between similar languages; this is due to the fact that languages with different orthographic basis may have different mapping between consonants and vowels. For example when translating English names to Arabic names many problems arise due to lexical difference. Firstly, Arabic deploys unwritten forms of short vowels in contrary with English names where short vowels are usually written. In such cases, Arabic short vowels (Fathah, Kasrah and Dammah) are being pronounced and should be used in the target language. Secondly, some Arabic consonants may be mapped to various English consonants, Examples: (س I s, c), (ب I b, p), (ك I k, c, ck), and others are mapped to more than one consonant, Ex. (ش I sh, ch), (th → ث ,ذ) which makes the problem a kind of many to many mapping task.

Finally, a general problem of Named Entities transliteration is that it is always preferable to produce the most commonly used form of the name.

In this paper we introduce a phrase based Arabic to English transliteration system to align Arabic substrings to English substrings based on parallel corpus of Aligned named Entities. A cascaded spelling suggested module is proposed to solve the problems that are beyond the phrase based transliteration limitations. The Spelling suggestion is applied over the phrase based transliteration system output to introduce best spelling correction for the transliterated name. This step makes our system more biased towards commonly used form of the name rather than the pure morphological representation.

1 Introduction

Named entities translation is strongly required in the field of Information retrieval (IR) as well as its usage in Machine translation and many other natural language processing applications.

In a statistical approach to statistical machine translation, given a foreign word F, we try to find the English word Ê that maximizes P(E\F). Using Bayes' rule, we can formulate the task as follows,

$$\hat{E} = \underset{E}{\text{argmax}} \frac{P(F\backslash E)*P(E)}{P(F)}$$

$$= \underset{E}{\text{argmax}}\ P(F\backslash E)*P(E)$$

This is known as the noisy channel approach to machine translation, which splits the problem into two sub-tasks. The translation model provides an estimate for the P(F\ E) for the foreign word F being a translation for the English word E, while the language model provides an estimate of the probability P(E) is an English word.

The phrase-based approach developed for statistical machine translation (Koehn et al., 2003) is designed to overcome the restrictions on many-to-many mappings in word-based translation models.

In this paper I introduce a statistical based learning of the characters mapping between source and target languages (Arabic to English), as well as vowel insertion in the target language. Then the most common transliteration is found by cascading a spelling correction module and a unigram language model.

Our approach is a two level solution, first of which is a statistical phrase based transliteration system which aligns Arabic to English substrings based on previously seen parallel aligned pairs of Arabic English names. Each Arabic name entry is transliterated then the output is cascaded to a Named entity spelling correction system, which suggests a better translation for the name in a most commonly used form. The spelling correction produces name suggestions with limited edit distance from the transliterated name and weights them according to the name frequency in a large monolingual English word list.

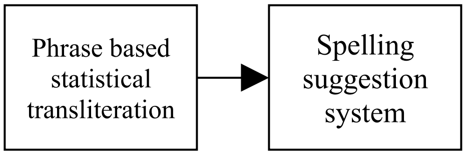
Section 2 is a brief description of related work, Section 3 includes a detailed description of our approach, and Section 4 described our experiments set up and the results will be reported in section 5. Finally in section 6 will describe our conclusions and future work.

2 Related Work

Most prior work in Arabic-related transliteration has been developed for the purpose of machine translation and for Arabic-English transliteration in particular. Arbabi (Arbabi et al., 1998) developed a hybrid neural network and knowledge-based system to generate multiple English spellings for Arabic person names. Stalls and Knight (Stalls and Knight, 1998) introduced an approach for Arabic-English back transliteration for names of English origin; this approach could only back transliterate to English the names that have an available pronunciation. Al-Onaizan and Knight (Al-Onaizan and Knight, 2002) proposed a spelling-based model which directly maps English letter sequences into Arabic letter sequences. Their model was trained on a small English Arabic names list without the need for English pronunciations. Although this method does not require the availability of English pronunciation, it has a serious limitation of not providing the most commonly used forms of the 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. Li et al. (2004) propose a letter-to-letter substring transliteration model for Chinese-English transliteration in an attempt to allow for the encoding of more contextual information. The model isolates individual mapping operations between training pairs, and then learns substring probabilities for sequences of these mapping operations. Ekbal et al. (2006) adapt this model to the transliteration of names from Bengali to English.

3 The Approach

Our approach is a cascaded Named entity Arabic to English translation system. Applying the phrase based statistical approach used in Machine translation on our problem, where the phrase is a sequence of n characters, trying to reach substring to substring of character sequence alignments.


Phrase based statistical transliteration
Spelling suggestion system

First we'll build a Hidden Marcov Model (HMM) phrase based transliteration system which aligns Arabic substrings to English character substrings, and then a cascaded stage of a spelling suggestion system suggests better name translation for the source name. In this framework we'll also need to introduce tools like ASpell

3.1 Transliteration

Training a Hidden Marcov Model (HMM) using a corpus of parallel English to Arabic named Entity pairs, to extract blocks of Arabic (A) characters sequence aligned to a sequence (E) of English characters in a procedure similar to what happens in Machine translation (MT). A feature table of English substrings mapped to Arabic substrings is considered as the translation model P(A\E). A language model is built over a large dictionary of English names to get the P(E). The translation model as well as the Language model is used by a beam search decoder to get best path of character sequence as a translation for the Arabic characters sequence.

In addition to the individual letters of the alphabet (unigrams), the target language (English) includes some character n-grams, for example: sh, th, ck, gh, that are phonetically aligned to a single Arabic character.

To generate English transliterations for an Arabic word, wa, For each segment, all possible transliterations, wb, are generated. Each word transliteration receives a score as follows which allows the transliterations to be ranked:

$P(wa \backslash we , wa \in A) = P(wa \backslash we) * P(wa \in A)$.

The experiments set up will be as follows:

i. In the first experiment English training words were re-segmented, that is, if a character was part of an n-gram mapped to single Arabic character, it was grouped with the other characters in the n-gram. If not, it was rendered separately.

For example:

g h a d a	➔	**gh** a d a
r i c h a r d	➔	r i **ch** a r d
b a r a c k	➔	b a r a **ck**
p h i l i p	➔	**ph** i l i p

This model included most commonly occurring English character sequences that function as units, such as ch, sh, th, ph, gh, kh, ck.

ii. Since Arabic short vowels are omitted from Arabic words, then corresponding English vowels as "a, e, o, u, i" are most probably aligned to "Null". This makes some problems for English names starting with vowels that are not aligned to Null. During decoding we faced vowel misalignment errors:

ابراهيم:	A b r A h y m	→	b r a h i m
امينة:	A m y n p	→	m e n

Thus, the second experiment investigates the effect of lowercasing versus true-casing the English target names Arabic to English character alignment especially English vowels alignment.

iii. Since Arabic to English transliteration is more difficult task than English to Arabic transliteration, we are going to find out the effect of reversing the training direction for getting a better Arabic English substring alignment.

iv. LM-Chunks:

- After the Arabic name is passed to the decoder as characters sequence, the decoder generates N best English characters sequence based on the translation model. These N best candidates are rescored according to a language model of valid English character sequences.

But the character based language model gives lower cost for shorter sequences, which may generate invalid character sequences as transliterations though they are not valid words.

So I built a dictionary of unigram chunks which scores the whole English transliteration candidates as unigrams in order to get more valid names.

Get best-N output from the decoder, and then rescore Top-N names according to Language Model Chunks.

3.2 Cascaded Spelling suggestion:

Some named entities were having different origin, Chinese, Japanese, Russian, etc… or had a de-facto translation which is beyond the scope of the data in the training data, for example "ميلوسفيتش" I "Milosevic".

This led us towards the idea of getting spelling variant suggestions which maybe not morphologically the quite suitable transliteration of the Arabic source, but it's the translation which people usually use.

This is quite useful with the problem of OOV (Out of Vocabulary) which appear in news with high frequency though they weren't well known at all in a previous time period.

4 Evaluation

In this section, I will describe the evaluation of our models on the task of Arabic-to-English transliteration.

4.1 Data and Resources

For our experiments, we used Arabic English named entity pairs for training the transliteration model, and for testing and development sets.

To train the language models, we simply needed a list of English names.

- Training data: About 70,000 parallel English Arabic names pairs (English is lower cased). Split these parallel names into characters.

- Language Model data: About 280,000 unique English names with counts in addition to all

English names in the training data were split into characters to allow Language Model check over the characters sequences during decoding.

- Large corpus of English unigrams: We have 13,653,070 unique words with frequency

  Lower cased: 4,022,239 words

  Upper cased: 5,592,717 words

This list was used as LM-chunks dictionary which ASpell suggest names from.

Also their frequency were used by a cost function to rescore the transliteration candidates to select most commonly used transliteration.

A test set of 1400 English-Arabic transliteration pairs contained no overlap with the set that was used to train the transliteration models, where each Arabic name has one, two or three English translation variants. The experiments report system performance over Best-1 and Best-3 English translations.

4.2 Evaluation Methodology

For each of the 1400 names pairs in the test set, each Arabic input was passed to the models, then English truth-1 and truth-3 of the test set were considered as gold standard transliteration for evaluating the system output.

Two separate tests were performed on the test set. In the first, the 1400 English words in the test set were added to the training data for the language models (the *seen* test), while in the second, all English words in the test set were removed from the language model's training data (the *unseen* test). Both tests were run on the same set of words to ensure that variations in performance for *seen* and *unseen* words were solely due to whether or not they appear in the language model (and not, for example, their language of origin).

4.3 System-1

As a baseline for our experiments, we trained a Hidden Marcov Model (HMM) using a corpus of parallel English to Arabic named Entity pairs, to extract blocks of Arabic (A) substring aligned to a sequence (E) of English substring.

In addition to the individual letters of the alphabet (unigrams), we merged all English bigrams that are phonetically aligned to a single Arabic character: sh, ch, th, gh, kh, ph, ck.

Find below the results for this experiment, using Lower cased English, True cased English and Vowel true cased English (true casing English names starting with vowels only, ex: Eman, Ibrahim, Ussama, …)

	Gold-1	Gold-(1-3)
Lower case	0.156	0.241
True case	0.165	0.267
Vowel only true case	0.164	0.262

Table (1)

Where Gold-1: is exact match precision over most frequent truth transliteration of the Arabic input.

Gold-(1-3): is exact match precision over one, two, till three most frequent transliterations of the Arabic input.

Since the true transliteration of an Arabic input may vary. For example: Mohamed, Muhammad, Mohammed are 3 valid transliterations for the Arabic input "محمد" , Romanized "mHmd".

We can find that either true casing names starting with vowels only or true casing all English names slightly improves the precision.

4.4 System-2: Reversing training direction

In this experiment we run the training as if English is our source language and Arabic is our target language, And knowing that Arabic short vowels are omitted from text. Training the model with English as the source language improves characters alignment and block extraction, because the HMM aligner prefers aligning a character to Null than aligning Null to characters.

Getting the English to Arabic substring mapping, you can reverse it back to get Arabic to English mappings used by the decoder.

This improved the alignment as well as the vowel insertion, the exact precision stepped up 5.5% as you can see in Table 2. We can also notice that true casing has no effect on the results.

	Gold-1	Gold-(1-3)
Lower case	22.1	34.9
True case	22.2	34.5

Table (2)

4.5 System-3: Using LM-Chunks

The difference between our problem and Machine translation is that in Machine translation we seek the best sequence of words that are nearest to the reference and are most expressing the meaning of the source stream. In named entities transliteration, though we mapped our problem to phrase based Machine translation, but we are seeking that the whole output characters sequence exactly matches one of the valid transliterations of the source stream because we are evaluation our system for true name transliterations it retrieved, more than measuring how close the mismatched outputs from the Gold reference.

For example:

“تشارلز I Charlez” ….. may be phonetically more close to the Arabic source, however the true transliteration should be “Charles”. The character based language model is not be capable of rising the cost of the wrong transliteration, so having a level of language modeling over the whole output English transliteration helps so much in improving the precision of output. We used a monolingual English dictionary, not necessary to be for names only, which is a more simple pool of data. After some filtering on the dictionary to make it more biased towards names, automatic filtering, and then we rescored the N-Best transliteration candidates with a cost function that rescores each candidate according to its transliteration cost and its cost in the chunks dictionary (or LM). This stepped our precision by 15% on the Best-1 matching and 20% on the Best-3 matching.

In Table (3) we reported the precision using LM chunks dictionary.

Then we added intentionally the truth of the test set to the LM chunks dictionary to know the effect of using a larger monolingual corpus in retrieving the Gold standard transliteration. This is be our “seen test”. The results are reported in table (3). It is noticeable that the precision for the seen test set stepped up 4-6% from the unseen test set.

	Gold-1	Gold-(1-3)
Unseen test	37.5	55.4
Seen test	41.4	61.3

Table(3)

Then we measured the average Levenshtein edit distance between the output transliteration and the Gold Best-1 truth transliteration. The results are included in table (4) for system-1, system-2, system-3, where the average name length = 7.279 characters

	Average edit distance (characters)
System-1	1.789
System-2	1.474
System-3	1.389

Table (4)

Referring to the results reported by Tarek Sherif and Grzegorz Kondrak ACL 2007, their experiments were held using seen test set. The precision on the best model they used was 70.0 %, outperforming my system. However, for the unseen test set they didn’t report the precision. Instead they reported a minimum average Levenshtein distance of 2.01 for the unseen test set that the Human transliteration average edit distance = 1.33. The results in table (4) indicate that the average edit distance of system-3 output from the Gold truth = 1.389 which is near from the average human edit distance reported by them.

4.6 Spelling suggestion transliteration

As the LM-chunks used in system-3 improved the precision of matching due to referring to valid English words in the chunks LM (or dictionary). We then tried to extend the idea of producing transliterations that are more biased towards what people use in real life as valid names or even valid words rather than producing character sequences that are phonetically close to the true transliteration but can not be use for retrieving the required Named entity during information retrieval, or even as a suitable transliteration in the output of machine translation. Since the LM-chunks used in system-3 weights the transliteration candidates based on their transliteration cost and their cost in the dictionary, then its output is restricted to what the trained transliteration model can produce and to the features the trainer could learn from the training data. But as new named entities are introduced on daily basis in newswire and many of which can be obtained easily in a monolingual large corpus but difficult to be found in a parallel corpus. Also many transliteration cases are back transliteration of non-English origin names. Chinese, Russian, German and named entities from other origins

are being written in English alphabet in a way biased towards their origin. For example names like “**Milosevic**” are phonetically pronounced in English as “**Milsosevich**” but it inherited the c→ ch mapping from its Russian origin.
Also for the Chinese “Xinhua” its pronounciation in English is near to “Shinhua” which makes its transliteration in Arabic as “شينهوا” phonetically too far from the true transliteration “Xinhua” and beyond the scope of transliteration candidates that the transliteration model can provide. Thus allowing a spelling suggestion module with a limited edit distance for such kind of named entities over the output of the transliteration model or over a simpler letter based transliteration transducer, may help in solving the problem.
I held limited experiments using the cascaded spelling correction, but only proved slight improvement. I will investigate it in further experiments since the cost function between the transliteration cost and the spelling correction cost is not settled yet before the submission deadline.

5 Conclusion and Future Work

In this paper we presented a substring based transliteration system, using the noisy channel approach being used in Machine translation, cascaded with Spelling suggestion system over a large English corpus to cover the difficult transliteration cases that are beyond the scope of the standard transliteration model.

We tested the effect of lower cased and true cased English character sequences, found that true casing improved the alignment and slightly increased the precision.

We also examined target reversing the training direction which proved to give better alignment, and improved the precision ~ 5-7 % in the unseen test.

We also tried using chunked Language Model to re-score the best N-transliterated candidates, where the transliteration precision jumped 15% on the Best-1 gold standard truth and 20% on the Best-3 gold standard truth.

Finally we explained the cascaded spelling suggestion module, and hopefully I will carry further experiments on it.

The overall precision of the systems are not satisfying to me. the baseline score was very low, will try the system on a more standardized test set in the future.

In the future I am planning to hold some experiments to filter out the generated phrase table, Arabic to English substring alignment, and try more decoding approachs.

6 Acknowledgement

I would like to thank Dr. Hany Hassan in IBM Cairo TDC for his helpful comments.

7 References

Yaser Al-Onaizan and Kevin Knight. 2002. Machine Transliteration of Names in Arabic Text. In Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages.

Tarek Sherif and Grzegorz Kondrak. 2007. Substring-Based Transliteration. In Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages.

N. AbdulJaleel and L. S. Larkey. 2003. Statistical transliteration for English-Arabic cross language information retrieval. In *CIKM*, pages 139–146.

A. Ekbal, S.K. Naskar, and S. Bandyopadhyay. 2006. A modified joint source-channel model for transliteration. In *COLING/ACL Poster Sessions*, pages 191–198.

Peter F. Brown, Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert L. Mercer. 1993. The Mathe-matics of Statistical Machine Translation: Parameter Estimation. Computational Linguistics, 19(2):263–311.

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

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

J. Zobel and P. Dart, Phonetic String Matching: Lessons from Information Retrieval. SIGIR Forum, special issue:166--172, 1996.

P. Koehn, F.J. Och, and D. Marcu. 2003. Statistical Phrase-Based Translation. *Proc. Of the Human Language Technology Conference*, *HLT-NAACL'2003*, May.