PRES LUNAM
Ecole Doctorale STIM
Sciences et Technologies de l'Information et Mathématiques

Spécialité : Informatique
Laboratoire : Laboratoire d'Informatique de l'Université du Maine (LIUM)
Equipe : Language and Speech Technology (LST)

Semi-supervised Transliteration Mining from Parallel Corpora

Aransa, Walid
Mél : walid.aransa@lium.univ-lemans.fr

Abstract: Transliteration is the process of writing a word (mainly proper noun) from one language in the alphabet of another language. This process requires mapping the pronunciation of the word from the source language to the closest possible pronunciation in the target language. In this paper we introduce a new semi-supervised transliteration mining method for parallel corpora. The method is mainly based on a new suggested Three Levels of Similarity (TLS) scores to extract the transliteration pairs. The first level calculates the similarity of of all vowel letters and consonants letters. The second level calculates the similarity of long vowels and vowel letters at beginning and end position of the words and consonants letters. The third level calculates the similarity of consonants letters only.

We applied our method on Arabic-English parallel corpora. We evaluated the extracted transliteration pairs using a statistical based transliteration system. This system is built using letters instead or words as tokens. The transliteration system achieves an accuracy of 0.46 and a mean F-score 0.88 when trained on transliteration pairs extracted from the parallel corpus. This shows that the proposed semi-supervised transliteration mining algorithm is effective and can be applied to other language pairs.

Keywords: *Arabic, English, Transliteration, Transliteration mining, Bitext, Corpora.*

1 Introduction

Transliteration is the process of writing a word (mainly proper noun) from one language in the alphabet of another language. This process requires mapping the pronunciation of the word from the original language to the closest possible pronunciation in the target language. Both the word and its transliteration are called a Transliteration Pair (TP). The automatic extraction of TPs from parallel or comparable corpora is called Transliteration Mining (TM). The transliteration pairs are important for many applications like Machine Translations (MT), machine transliteration, cross language information retrieval (IR) and Name Entity Recognition (NER). For example, in MT, TM can be used to improve the word alignments, or to train a system to transliterate proper nouns in out-of-vocabulary (OOV) words. In machine transliteration, the obtained TPs are used to train statistical transliteration system, while in IR, it is used to enrich the search results with orthographical variations.

Recently, TM has gained considerable attention from the research community. There are several methods to perform TM: supervised, unsupervised and semi-supervised. Also, some TM researches focus on parallel corpora and others on comparable corpora. In this paper we will focus on semi-supervised method on parallel corpora.

The paper is organized as follows: the next section presents related work, followed by a description of the TM algorithm when using parallel corpora. The paper concludes with a discussion of the perspectives of this work.

2 Related work

There are several methods to perform TM, supervised, unsupervised and semi-supervised, some TM researches focus on parallel corpora and others on comparable corpora. [1] uses variant of the SOUNDEX methods and n-grams to improve precision and recall of name matching in the context of transliterated Arabic name search. Original, SOUNDEX was developed by [2]. Another method proposed by [1] reduces the orthographical variations by 30% using SOUNDEX improved precision slightly but they observed a decrease in recall.

For transliteration research, [3] uses two algorithms based on sound and spelling mappings using finite state machines to perform the transliteration of Arabic names. [4] presents DirecTL, a language inde-

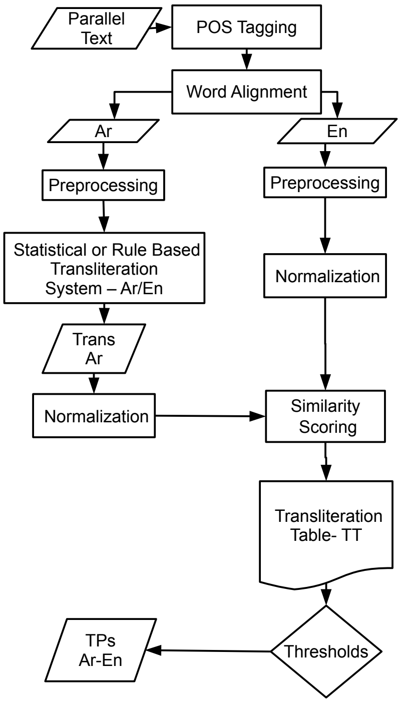

Figure 1: Extracting TPs from parallel corpora

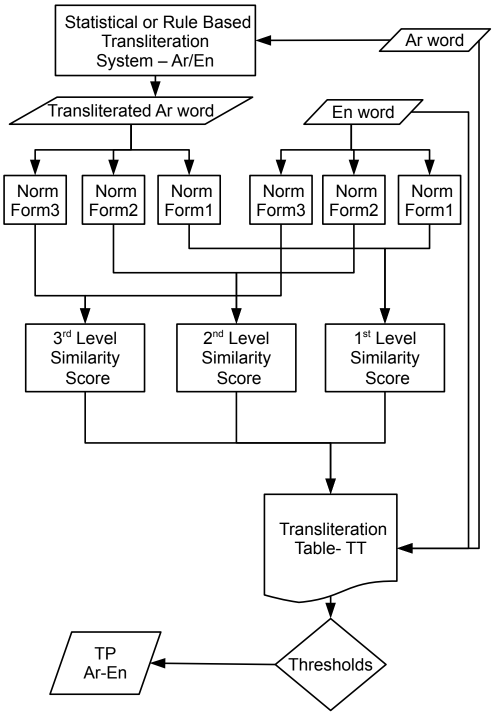
Statistical or Rule Based Transliteration System – Ar/En
Ar word
Transliterated Ar word
En word
Norm Form3
Norm Form2
Norm Form1
Norm Form3
Norm Form2
Norm Form1
3rd Level Similarity Score
2nd Level Similarity Score
1st Level Similarity Score
Transliteration Table- TT
Thresholds
TP Ar-En

Figure 2: Calculating the three levels of similarity scores

pendent approach to transliteration. DirecTL is based on an online discriminative sequence prediction model that employes EM-based many-to-many unsupervised alignment between target and source.

3 Transliteration mining using parallel corpora - semi-supervised

3.1 TM algorithm for parallel corpora

The algorithm as shown in Figure 1 is designed to compare two aligned words and detect the words which are transliteration of each other. We developed the following TM algorithm:

(1) First, the parallel corpus is tagged using a part-of-speech (POS) tagger. We used Stanford POS tagger [5] for English and Mada/Tokan for Arabic POS tagging.

(2) Then, we align the tagged bitext using Giza++ [?], using the source/target alignment file, remove all aligned word pairs with POS tags other than noun (NN) or proper noun (PNN) tags and remove all English words starting with lower-case letters. Words which have most lowest alignment scores are removed (about 5% from the total number of aligned word pairs).

(3) After that removing the POS tags from Arabic and English words.

(4) Then, transliterate the Arabic word A into English using a rule based transliteration system (or a previously trained statistical based transliteration system).

(5) Normalize the transliteration of Arabic word A_t as well as the English word to $Norm_1$, $Norm_2$ and $Norm_3$ as explained in section 3.2. The objective of the normalization is folding English letters with similar phonetic to the same letter or symbol.

(6) For each aligned Arabic transliterated word A_t and English word E, use their normalized forms to calculate the three levels of similarity scores which we store in a transliteration table (TT).

(7) Extract TPs from the TT by applying a threshold on the three levels similarity scores. We selected the thresholds using empirical method shown in section 3.4.1.

3.2 English normalization and three levels similarity scores for TM

As shown in Figure 2, we developed a three normalization functions which can be used to normalize the Arabic transliterated word and English word to be more comparable to each other phonically. These normalized forms are used to calculate the similarity between the transliterated word and the English word based on three levels of similarity. The first level calculates the similarity of all vowel letters and consonants letters. The second level calculates the similarity of long vowels and vowel letters at beginning and end position of the words as well as consonants letters. The third level calculates the similarity of consonants letters only. The details of each normalization function as following:

(1) $Norm_1$ normalization function: Normalize the transliteration of Arabic word as well as the English word. The objective of the normalization is folding English letters with similar phonetic to one letter or symbol. In $Norm_1$, all letters are converted to lower case, phonically equivalent consonants and vowels are folded to one letter (e.g. p and b are normalized to b, v and f are normalized to f, i and e are normalized to e), double consonants are replaced by one letter, and finally a hyphen "-" is inserted after the initial two letters "al" if it is not already followed by it.

(2) $Norm_2$ normalization function: Using $Norm_1$ output, double vowels are replaced by one similar upper-case letter (i.e. ee is normalized to E), remove non-initial and non-final vowels only if not followed by vowel or not preceded by vowel.

(3) $Norm_3$ normalization function: Using $Norm_2$, hyphen "-" and vowels are removed.

Hence, for each Arabic word A and English word E. if A_t is the transliteration of A into English, we can calculate the following three levels similarity scores while i=1,2,3

$$TLS_i = \frac{Levenshtein(Norm_i(A_t), Norm_i(E))}{|Norm_i(E)|} \tag{1}$$

In this formula, Levenshtein function is the edit distance between the two words, which is the number of single-character edits required to change the first word into the second one.

3.3 Transliteration system for TM evaluation

The transliteration system is built using the moses toolkit [7]. We train a letter-based SMT system on the list of TPs extracted using our TM algorithm explained in section 3.1. The distortion limit is set to 0 to disable any reordering. The transliteration system should be able to learn the proper letter mapping using the alignment of the letters, and hence be able to generate the possible transliterations of a name written in the source language script using the learned mapping rules into a name written in the target language script.

3.4 Experiments and evaluation

3.4.1 Three levels similarity scores thresholds selections

Several systems were trained to evaluate the best thresholds to be used in our experiments. The experiments show that the best thresholds for 3-scores on tuning set are (TLS_3, TLS_2, TLS_1)=(0, 0.39, 0.49). The thresholds are highly dependent on the normalization functions $Norm_1$, $Norm_2$ and $Norm_3$, so changing the normalization functions will require a re-selection of the three thresholds. The scores of the TuningSet with different thresholds are mentioned in Table 1. Table 2 lists the systems with the TLS scores' thresholds used to select data to train each one.

System(*)	ACC	Mean F-Score	MRR	MAP_{ref}
SYS013 TPs=9167	0.43545	0.87940	0.54188	0.43545
SYS023 TPs=9070	0.44159	0.87860	0.54862	0.44160
SYS034 TPs=10529	0.44774	0.88226	0.55012	0.44774
SYS134 TPs=10529	0.43647	0.88042	0.54220	0.43647

Table 1: *Tuning set results with different thresholds*

System	TLS_3	TLS_2	TLS_1
SYS013	0	0.19	0.39
SYS023	0	0.29	0.39
SYS034	0	0.39	0.49
SYS134	0.19	0.39	0.49

Table 2: *TLS scores' thresholds used for each system*

3.4.2 Results

Using three levels similarity scores thresholds=(0, 0.29, 0.39) as explained in section 3.4.1, the total number of extracted TPs is 12988. Table 3 shows the percentage of extracted TPs as a function of the number of aligned words in the parallel text and the number of aligned words with an NNP/NN POS tag.

Data	Number	Extracted
Data	of Words	TPs %
Bitext-Arabic	3.8M	0.24 %
Bitext-English	4.4M	0.21 %
List of aligned words	1249167	0.73 %
List of aligned NN*	161811	5.6 %

Table 3: *Extracted TPs rate*

System	ACC	Mean	MRR	MAP_{ref}
	ACC	F-Score		
TuningSet	0.5000	0.8958	0.6117	0.5000
TestSet	0.4616	0.8841	0.5822	0.4616

Table 4: *TuningSet and TestSet scores*

In Table 4, we list the transliteration system results. We report the scores for both TuningSet and TestSet. Both TuningSet and TestSet have not seen before in the training data.

4 Conclusions

In this paper we introduce a new semi-supervised transliteration mining method for parallel corpora. The method is mainly based on new suggested Three Levels of Similarity (TLS) scores to extract the transliteration pairs. The transliteration system trained on the transliteration pairs extracted from the parallel corpus achieves an accuracy of 0.46 and a mean F-score of 0.88 on the test set of unseen Arabic names. This shows that the proposed semi-supervised transliteration mining algorithm is effective and can be applied to other language pairs.

References

[1] David Holmes, Samsum Kashfi, and Syed Uzair Aqeel. Transliterated arabic name search. In M. H. Hamza, editor, *Communications, Internet, and Information Technology*, pages 267–273. IASTED/ACTA Press, 2004.

[2] Robert Russell. Specifications of letters. US patent number 1,261,167, 1918.

[3] Yaser Al-Onaizan and Kevin Knight. Machine transliteration of names in arabic text. In *Proceedings of the ACL-02 workshop on Computational approaches to semitic languages*, SEMITIC '02, pages 1–13. Association for Computational Linguistics, 2002.

[4] Sittichai Jiampojamarn, Aditya Bhargava, Qing Dou, Kenneth Dwyer, and Grzegorz Kondrak. Directl: a language-independent approach to transliteration. In *Proceedings of the 2009 Named Entities Workshop: Shared Task on Transliteration*, NEWS '09, pages 28–31. Association for Computational Linguistics, 2009.

[5] Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 173–180. Association for Computational Linguistics, 2003.

[6] Franz Josef Och and Hermann Ney. A systematic comparison of various statistical alignment models. *Comput. Linguist.*, 29(1):19–51, March 2003.

[7] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. Moses: open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, pages 177–180. Association for Computational Linguistics, 2007.