Semi-supervised Transliteration Mining from Parallel and Comparable Corpora

Walid Aransa, Holger Schwenk, Loic Barrault

LIUM - University of Le Mans, France
firstname.lastname@lium.univ-lemans.fr

Dec 7th 2012
IWSLT 2012, December 6-7, 2012, Hong Kong

Outline

1. Introduction
   - Transliteration
   - Transliteration challenges
   - Transliteration mining
2. Related work
3. Transliteration mining using parallel corpora - semi-supervised
4. Transliteration mining using comparable corpora - semi-supervised
5. Conclusion

Introduction

- Transliteration is the process of writing a word (mainly proper noun) from one language in the alphabet of another language.
- It requires mapping the pronunciation of the word from the original language to the closest possible pronunciation in the target language
- The word and its transliteration are called a Transliteration Pair (TP)

Transliteration applications

- Machine Translations:
  improve the word alignments, OOV
- Machine Transliterations:
  train statistical transliteration system
- Cross language Information Retrieval (IR):
  enrich the search results with orthographical variations
- Name Entity Recognition (NER)

Transliteration challenges

- Examples: Transliteration from Arabic into English
- Some Arabic letters have no phonically equivalent letters in English (e.g. ض and ط)
- Some English letters do not have phonically equivalent letters in Arabic (e.g. v)
- Missing of short vowels (i.e. diacritics) in the Arabic text
- Some Arabic letters can be mapped to any letter from a group of phonically close English letters (e.g. ب to "p or b")
- Some Arabic letters can be mapped to a sequence of English letters (e.g. خ to 'kh')

Transliteration challenges - Cont

- Tokenization challenges: the Arabic name is concatenated to clitics like:
  - Preposition بـ
  - Conjunction و
  - Both together (e.g. وبـ)
- Transliteration types:
  - Forward: name is transliterated from its original language to another language

    Example: Arabic origin name "محمد" -> "Mohamed"
  - Backward: the transliterated names are transliterated back to the origin names in its original language

    Example: "بوش" -> "Bush"

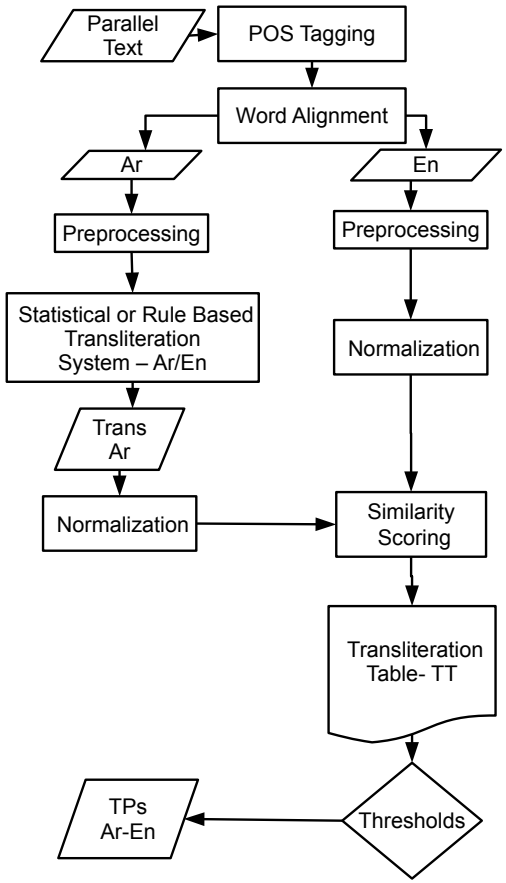
Transliteration mining (TM)

- The automatic extraction of TPs from parallel or comparable corpora is called Transliteration Mining (TM)
- Several methods to perform TM:
  - Supervised
  - Unsupervised
  - Semi-supervised
- Some TM researches focus:
  - Parallel corpora
  - Comparable corpora

Related work

- (Holmes et al., 2004) uses variant of the SOUNDEX methods and n-grams
- It improves precision and recall of name matching in the context of transliterated Arabic name search.
- (Darwish, 2010) presents two methods for improving TM, phonetic conflation of letters and iterative training of a transliteration model.
- The first method is an improved SOUNDEX phonetic algorithm. They propose SOUNDEX like conflation scheme to improve the recall and F-measure.
- Also iterative training method was presented that improves the recall but decreases the precision.

TM using parallel corpora - semi-supervised



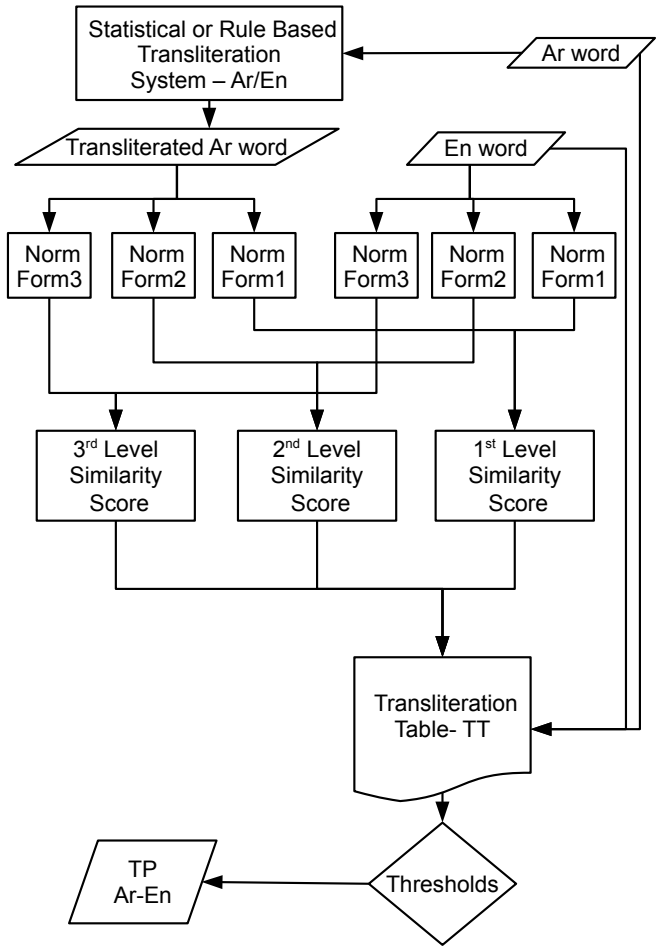
TM algorithm for parallel corpora

(1) The parallel corpus is tagged using a part-of-speech (POS) tagger. We used Stanford POS tagger for English and Mada/Tokan for Arabic POS tagging.
(2) 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) Remove the POS tags from Arabic and English words.
(4) Transliterate the Arabic word A into English using a rule based transliteration system (or a previously trained statistical based transliteration system).

TM algorithm for parallel corpora - Cont

(5) Normalize the transliteration of Arabic word A_t as well as the English word to $Norm_1$, $Norm_2$ and $Norm_3$ as will be explained. 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 later.

Calculating the three levels of similarity scores



Calculating the three levels of similarity scores - Cont

(1) $Norm_1$ normalization function: folding English letters with similar phonetic to one letter or symbol.

- lower cased
- 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
- hyphen "-" is inserted after the initial two letters "al" which is the transliteration of Arabic article "ال"

Calculating the three levels of similarity scores - Cont

(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 $Norm_2$ output, hyphen "-" and vowels are removed.

Calculating the three levels of similarity scores - Cont

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

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

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.

Experiment and evaluation

- Corpora:
  - Arabic/English parallel corpus
  - 3.8 million Arabic words
  - 4.4 million English words
- The extracted TPs are used as training data
- We built the TuningSet and TestSet from the extracted TPs

Experiment and evaluation

- The extracted TPs are divided into three parts:
  1. Training data set. The size of the training data is variable based on the selected three levels thresholds (9070 pairs to 10529 TPs)
  2. Tuning data set (1k TPs)
  3. Test data set. (1k TPs)
- All occurrences of words in the TuningSet or TestSet are removed from the training data.

Three levels similarity scores thresholds selection

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

Table: *Tuning set results with different thresholds*

Three levels similarity scores thresholds selection

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

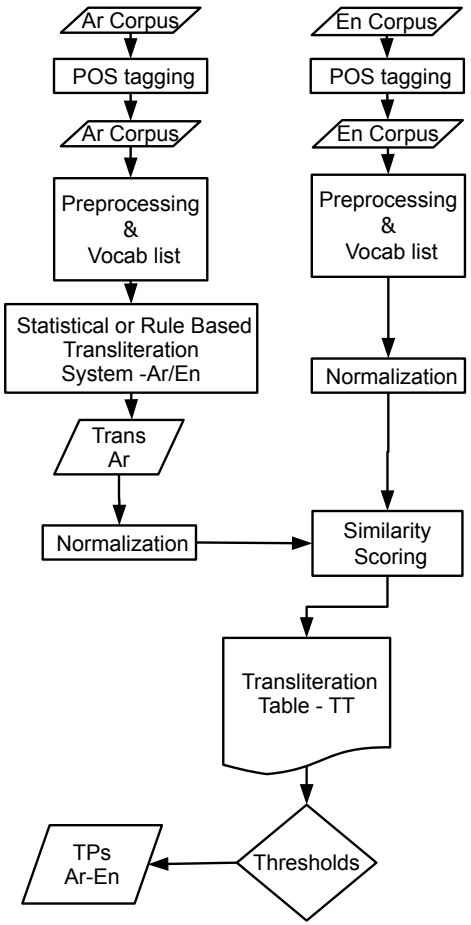
Table: *TLS scores' thresholds used for each system*

TM algorithm for parallel corpora - Results

System	ACC	Mean F-Score	MRR	MAP_{ref}
TuningSet	0.50000	0.89589	0.61178	0.5000
TestSet	0.46162	0.88412	0.58221	0.4616

Table: *TuningSet and TestSet scores*

TM using comparable corpora - semi-supervised



TM algorithm for comparable corpora

(1) Each monolingual corpus is tagged using part-of-speech (POS) tagger. We used Stanford POS tagger for English and Mada/Tokan for Arabic POS tagging.
(2) Remove all words with POS tags other than noun (NN) or proper noun (PNN) tags and from the remaining words, remove all English words starts with lower-case letters.
(3) Removing the POS tags from source text and target text.

TM algorithm for comparable corpora

(4) Derive two unique words lists (LIST_SRC and LIST_TRG) from both source and target texts.
(5) Transliterate source words list (LIST_SRC) into target language (LIST_SRC_TRANS) using rule based transliteration system (or previously created statistical based transliteration system).
(6) Normalize the transliteration of source words list as well as the English words list to the three normalized forms $Norm_1$, $Norm_2$ and $Norm_3$ as shown before. The objective of the normalization is folding English letters with similar or close phonetic to same letter or symbol.

TM algorithm for comparable corpora

(7) Using the normalized values, for each transliterated word in the source language list WORD_AR_TRANS and target language word WORD_EN, calculate the 3-similarity scores between them which are stored in the transliteration table (TT).
(8) Extract TPs from the TT by applying a selected three thresholds on the three levels similarity scores.

Experiment and evaluation

- Corpora:
  - Arabic Gigaword corpus (about 270.3 million Arabic words using only XIN, AFP and NYT parts)
  - English Gigaword corpus (roughly 1470.3 million English words using only XIN, AFP and NYT parts)
- The extracted TPs are used as training data. We used the same TuningSet and TestSet extracted from parallel corpus
- As before, all occurrences of words in the TuningSet or TestSet were removed from the training data.

Three levels similarity scores thresholds selections

System	ACC	Mean F-Score	MRR	MAP_{ref}
GSYS013 TPs=1.63M	0.30021	0.83973	0.40807	0.30021
GSYS023 TPs=1.96M	0.30021	0.84001	0.40817	0.30021

Table: *Tuning set results with different thresholds*

System(*)	TLS_3	TLS_2	TLS_1
GSYS013	0	0.19	0.39
GSYS023	0	0.29	0.39

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

TM algorithm for comparable corpora - Results

System	ACC	Mean F-Score	MRR	MAP_{ref}
TuningSet	0.30021	0.84001	0.40817	0.30021
TestSet	0.27329	0.83345	0.39788	0.27329

Table: *TuningSet and TestSet scores*

Conclusion

- We applied the Three Levels of Similarity (TLS) scores to extract the transliteration pairs.
- Applied the translation mining approach on two Arabic and English Parallel and Comparable corpora.
- The transliteration system trained on the transliteration pairs extracted from the parallel corpus achieves an accuracy of 0.50 and a mean F-score of 0.84 on the tune set of unseen Arabic names.
- The system trained on transliteration pairs extracted from comparable corpora achieves an accuracy of 0.30 and a mean F-score of 0.84 on the tune 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.

lium

Backup slides

Evaluation metrics

We used the de-facto standard metrics from ACL Name Entity Workshop (NEWS) (Zhang et al.,2012): ACC, mean F-Score, MRR, and MAP_{ref}. Here is a short description of each metric:

- ACC=Word Accuracy in Top-1, also known as Word Error Rate. It measures correctness of the first transliteration candidate in the candidate list produced by a transliteration system.
- F-Score= Fuzziness in Top-1. The mean F-score measures how different, on average, the top transliteration candidate is from its closest reference.

Evaluation metrics

- MRR=Mean Reciprocal Rank measures traditional MRR for any right answer produced by the system, among the candidates.
- MAP_{ref} tightly measures the precision in the n-best candidates for the i-th source name, for which reference transliterations are available.

Related work

- SOUNDEX was developed by (Russell, 1918) which is an algorithm used for indexing names by sound as pronounced in English.
- The SOUNDEX code for a name consists of a letter followed by three numerical digits: the letter is the first letter of the name, and the digits encode the remaining consonants.
- The method proposed by them reduces the orthographical variations by 30% using SOUNDEX improved precision slightly but they observed a decrease in recall.

Data for language model training

- LM1 is obtained from the English Gigaword corpus (using only XIN, AFP and NYT parts) by extracting a list of proper names using the Stanford name entity recognizer (NER).
- The second resource (LM2) is the English part of the extracted TPs.

System	ACC	Mean F-Score	MRR	MAP_{ref}
LM1	0.43750	0.88160	0.54787	0.43750
LM2	0.44159	0.87860	0.54862	0.44160

Table: *LM1 vs. LM2*

Segmentations techniques results

- Tried two English side segmentation techniques:
  - Individual letters
  - Advanced segmentation using group of letters that form one phonetic sound in one segment (e.g. ph, ch, sh, etc)

System	ACC	Mean F-Score	MRR	MAP_{ref}
One letter	0.47951	0.89248	0.59226	0.47951
1-2 letters	0.50000	0.89589	0.61178	0.5000

Table: *One letter segmentation vs. Advanced segmentation*

Pronunciation observations for Ar-En transliteration

- In most cases, we can sort the letter's impact on transliteration from low to high as following:
  - Phonically similar vowels have low impact.
  - Phonically dissimilar vowels have medium impact.
  - Consonants letters have significant impact.
- The double vowels produce long vowel sound have more impact on the pronunciation of the English word.

Pronunciation observations for Ar-En transliteration - Cont

- The sequence of two or more different vowel letters, has a special pronunciation which has more impact on the pronunciation of the English word.
- The vowel at the initial position or at the final position in the word has significant impact on the pronunciation. The same applies for consonants (e.g. consider the following two names: Adham, Samy)

TM algorithm for parallel corpora - Results

Data	Number of Words	Extracted 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	5.60 %

Table: *Extracted TPs rate*

TM algorithm for comparable corpora - Results

Data	Number of Words	Extracted TPs %
Arabic Gigaword	270.3 M	0.73%
Arabic Gigaword NN*	18.7 M	10.48%
English Gigaword	1470.3 M	0.13%
English Gigaword NN*	8.1 M	24.20%

Table: *Extracted TPs rate*

System	ACC	Mean F-Score	MRR	MAP_{ref}
TuningSet	0.30021	0.84001	0.40817	0.30021
TestSet	0.27329	0.83345	0.39788	0.27329

Table: *TuningSet and TestSet scores*

lium