English-Arabic Transliteration

MOHAMED ABDEL FATTAH [1], FUJI REN [1, 2]
[1]. Faculty of Engineering, University of Tokushima
2-1 Minamijosanjima
Tokushima, 770-8506, JAPAN
[2]. School of Information Engineering, Beijing University of Posts & Telecommunications
Beijing, 100088, CHINA

Abstract: - Proper nouns may be considered as the most important query words in information retrieval. If the two languages use the same alphabet, the same proper nouns can be found in either language. However, if the two languages use different alphabets, the names must be transliterated. Short vowels are not usually marked on the Arabic words in almost all Arabic documents (except very important documents like the Muslim and Christian holy books). Moreover, most of Arabic words have a syllable of consonant-vowel (CV) which means that most of the Arabic words contain short or long vowel between two successive consonant letters. That makes it difficult to create English- Arabic transliteration pairs since some English letters may not be matched with any Romanized Arabic letter. In the present study, we present different approaches for transliteration proper noun pair's extraction from parallel corpora based on different similarity measures between the English and Romanized Arabic proper nouns under consideration. The strength of our new system is that it works well for low-frequency proper noun pairs. We evaluate the presented new approaches using two different English- Arabic parallel corpora. Most of our results outperform previously published results in terms of precision, recall and F- Measure.

Key-Words: - Machine transliteration, Parallel corpora, Cross-language information retrieval.

1 Introduction

Recently, much research has been done on machine transliteration for many language pairs, such as English/Arabic [1,2], English/Chinese [3], English/ Japanese [4], and English/Korean [5]. Most of the above approaches require a pronunciation dictionary for converting a source word into a sequence of pronunciations. However, words with unknown pronunciations may cause problems for transliteration. On the other hand, much research has focused on the study of automatic bilingual lexicon construction based on bilingual corpora. Proper nouns and corresponding transliterations can often be found in parallel corpora or topic-related bilingual comparable corpora. However, many methods dealt with this problem based on the frequencies of words appearing in corpora, an approach which cannot be effectively applied to low-frequency words, such as transliterated words [6]. Fung, used different approaches to create translation pairs from parallel and comparable corpora [7, 8, 9]. For instance, in[7], she presented a pattern matching method for compiling a bilingual lexicon of nouns and proper nouns from unaligned, noisy parallel texts of Asian/Indo-European language pairs. Although the simplicity of the used approach the recall was very small. On the other hand, Fattah et al., [6] presented two algorithms and their combination to automatically extract an English/Arabic bilingual dictionary from parallel texts that exist in the Internet archive after using an Arabic light stemmer as a preprocessing step. Both Fung and Fattah approaches do not require pronunciation dictionary for converting a source word into a sequence of pronunciations and they give reasonable results. Therefore, we have exploited the pattern matching method of Fung, [7] and Fattah's approach to extract transliteration pairs from English – Arabic parallel corpus and we used them as base line methods.

1.1 Pattern matching approach

In pattern matching approach, tagging information of one language is used. Word frequency and position information for high and low frequency words are represented in two different vector forms for pattern matching.

1.2 Combination of algorithms 1 & 2 by Fattah et al. [6]

The First algorithm of Fattah et al., uses a similarity metric S(a, e) between words in Arabic language (A) and words in English language (E) based on statistical co-occurrence and the frequency of each Arabic and English word. Then, it computes the association scores for a set of translation pairs (a, e) ∈ (A, E). Depending on a certain threshold, the translation pairs whose association score exceeds this threshold become the entries in the translation lexicon. The second algorithm of Fattah et al., is based on statistical co-occurrence and the frequency of each Arabic and English word too. However, it can extract translation pairs from two sentence pairs only. This algorithm can capture dependencies between groups of words to get word / phrase translation pair which was the problem of many statistical approaches like the first algorithm. Using the first algorithm, we can achieve high precision with low recall. However it is difficult to handle the translation of compound nouns. The second algorithm does not have the disadvantages of the first algorithm since it can handle the translation of compound nouns. Moreover the precision and recall are higher than that of the first algorithm. However the processing time required for the second algorithm is higher than that of the first one. This led us to use a certain combination of algorithm 1 and algorithm 2 to gain the advantages of both of them and avoid the disadvantages as much as possible.

2 The proposed English-Arabic proper noun transliteration pairs creation approach

The proposed English-Arabic proper noun transliteration pair's creation system extracts all proper nouns from the English sentence using the CLAWS4 POS tagger (http://www.comp.lancs.ac.uk/computing/research/ucrel/claws/trial.html). It also extracts all proper nouns from the associated Arabic sentence using the Buckwalter Arabic Morphological Analyzer Version 1.0. All the Arabic proper nouns are romanized using the table in (http://archimedes.fas.harvard.edu/mdh/arabic/arabic-loc.pdf). The similarity (based on different similarity measures as will be illustrated in the coming sections) between every English and Romanized Arabic proper noun pair is measured. The English- Arabic proper noun pair which has similarity score above certain threshold (th) is extracted. The system repeats this step for all English and Arabic proper nouns that exist in the sentence pair. The system applies the previous steps on all remaining sentence pairs to create all possible transliteration pairs available in the corpus under consideration. The following pseudo-code illustrates the previously mentioned methodology steps.

The Methodology Pseudo-Code

Set ie = ia = n = 1.
E: Extract proper nouns of English_Sentence(n) and Arabic_Sentence(n)
R: Romanize Arabic_Proper_Noun(ia)
Score = SIM (Romanized_Arabic_Proper_Noun(ia), English_Proper_Noun(ie))
If Score >= th
Copy Arabic_Proper_Noun(ia) & English_Proper_Noun(ie)with the score value in a file
End
ia= ia +1
if ia <= na
GOTO R
End
Set ia = 1 & ie = ie + 1
if ie <= ne
GOTO R
End
Set ia = ie = 1 & n = n + 1
If n <= N
GOTO E
End

Where, *na* and *ne* are the total number of Arabic and English proper nouns in the Arabic and English sentence number *n* respectively. "*th*" is a predefined threshold. *SIM* is the similarity measure that will be defined in the following sections. *N* is the total number of English- Arabic sentence pairs. The Arabic proper noun consonant and long vowel letters are romanized according to the table in (http://archimedes.fas.harvard.edu/mdh/arabic/arabic-loc.pdf).

The following sections describe Dice's Similarity Coefficient besides two proposed different similarity measures to measure the similarity between English and Romanized Arabic proper nouns.

2.1 Dice's Similarity Coefficient

Dice's similarity coefficient was originally developed in the field of biology to describe the degree of similarity between two species of plant according to the number of features (such as hairy stems) that they had in common. McEnery & Oakes [10] used Dice's similarity coefficient to describe the degree of similarity between a word in one language and its translation in another.

2.2 Similarity measure 1 (SIM1)

Most of Arabic words have a syllable of CV. Most of the Arabic words contain a short or long vowel between two consonant letters. Take the Arabic word "محمد" "mohammad" as an example. The short vowels 'o', 'a' and 'a' existed between the consonants "m, h", "h, m" and "m, d" respectively. Moreover the short vowels do not appear on the Arabic words in almost all Arabic documents. Hence, the Dice's approach to measure similarity between English- Arabic transliteration pairs does not work well. We have decided to use our proposed similarity measure called "SIM1". Using SIM1, the system specifies the similarity score between the English and Romanized Arabic words by matching the consonant characters of the English word with the Romanized Arabic characters. The system excludes the effect of vowel between two successive consonants and the repeated consonants by using the following algorithm to specify SIM1:

Set SIM1 = 0
Set ia = ie = 0
R: Read the Romanized Arabic character(ia)
Read the English character(ie)
If (the Romanized Arabic character(ia) = the English character(ie))
SIM1 = SIM1 + 1
End
Else ie = ie + 1 & Read the English character(ie)
If (the Romanized Arabic character(ia) = the English character(ie))
SIM1 = SIM1 + 1
End
Else If(English character(ie - 2)= English character(ie - 1)))
ie = ie+1 & Read the English character(ie)
If (the Romanized Arabic character(ia) = the English character(ie))
SIM1 = SIM1 + 1
End
End
ia = ia + 1 & ie = ie + 1
if (ia < Length (Romanized Arabic word))
GOTO R
End
SIM1 = SIM1/(max_Length(Romanized Arabic word, English word))

2.3 Similarity measure 2 (SIM2)

Using SIM2, the system restricts the extracted transliteration pairs only to the pairs that have all Romanized Arabic characters matched with some or all English proper noun characters to increase the precision. We achieve that by modifying the algorithm mentioned in section 2.2 to set the similarity score to zero if any Romanized Arabic character does not match with any English character. Therefore, for using any threshold value (th) > 0, the transliteration pairs that do not have all Romanized Arabic characters matched with some or all English proper noun characters are excluded.

3 Experimental Results

We have applied our transliteration techniques on the "Arabic English Parallel News Text Part 1", Linguistic Data Consortium (LDC) catalog number LDC2004T18 and ISBN 1-58563-310-0. This corpus contains Arabic news stories and their English translations LDC collected via Ummah Press Service from January 2001 to September 2004. It totals 8,439 story pairs, 68,685 sentence pairs, 2M Arabic words and 2.5M English words (http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp ?catalogId=LDC2004T18). This corpus contains 8827 English proper nouns specified by using the CLAWS4 POS tagger.

3.1 Experimental results using Pattern matching approach

We treat the transliteration compilation problem as a pattern matching problem in [7]. We applied the approach on the English-Arabic corpus. We achieved precision and recall of 68.3% 67.4% respectively for the best matched pairs. We also achieved precision

and recall of 71.6% and 69.7% respectively for the top three Arabic transliterations for an English proper noun respectively. Then we did little modification in the first step of the approach to increase precision. In the first step of the algorithm, we did not tag the English half of the parallel text only but we also tagged the Arabic half in order to restrict the matching process to as few words as possible to increase precision. We achieved precision and recall of 71.4% 66.5% respectively for the best matched pairs. We also achieved precision and recall of 73.8% and 68.2% respectively for the top three Arabic transliterations for an English proper noun respectively. We found that many mistaken transliterations resulted from insufficient data.

3.2 Experimental results using Algorithms 1 & 2 combination of Fattah et al., [6]

We have applied the Algorithms 1 & 2 combination of Fattah et al., on the Arabic English Parallel News Text Part 1 corpus after stemming and preprocessing steps that were mentioned in Fattah et al. We achieved precision of 89.3% and recall of 74.6%. The precision is better than that of Fattah et al. However, the recall is deteriorated. The reason of low recall is that many proper nouns are compound nouns such as the Arabic proper name "عبد القادر" and the English transliteration of it "Abdel Qader". The first algorithm of Fattah et al., fails to capture dependencies between groups of words and the second algorithm of Fattah et al., can extract the word and phrase translation of one word only. Hence, the Algorithms 1 & 2 combination failed to extract the compound nouns such as "عبد القادر" and "Abdel Qader" which deteriorates the recall.

3.3 Experimental results using Dice's Similarity Coefficient

We have applied the proposed approach using Dice's Similarity Coefficient on the English-Arabic parallel corpora to extract all possible transliteration pairs.

Apply the previous Pseudo-Code on the English-Arabic corpora, and use Dice's Similarity Coefficient to measure the similarity between the English proper noun and the Romanized Arabic proper noun. We extract all transliteration pair of similarity scores that exceed the threshold "th" value. Table 1 show the precision, recall and the harmonic mean of precision and recall (F-Measure) for the transliteration pairs extracted as a function of the threshold "th".

Table 1: The results using Dice's Similarity Coefficient

th	1.0	0.9	0.8	0.7
Precision	100%	100%	95.9%	86.7%
Recall	2.3%	2.3%	6.2%	15.3%
F-Measure	4.5%	4.5%	11.6%	26.0%
th	0.6	0.5	0.3	0.0
Precision	72.1%	61.3%	42.8%	24.2%
Recall	22.6%	28.7%	36.5%	98.1%
F-Measure	34.4%	39.1%	39.4%	38.8%

As shown in table 1, the precision is high especially for "th >= 0.7" however, the recall is too small in all values of "th" except "th" = 0.0. As "th" decreases, the precision decreases and the recall increases as well. For "th" = 0.0, the system creates all possible transliteration pair combinations included in a sentence pair. Hence, at "th" = 0.0, the precision is minimum while the recall is maximum. At "th" = 0.0, the recall is not 100% since there is some error caused by the English tagger and the Arabic morphological analyzer tools. The precision and recall at "th" = 0.0 are fixed for any similarity measure used. From table 1 we can see that the recall is low in general since the Dice's approach is not the best way to measure similarity between an English proper noun and the Romanized Arabic proper noun. That is because the Arabic diacritics are not typed in most of Arabic documents. Take the following example to illustrate this point:

The Arabic proper noun "محمد = Mohammad" is written in a very few documents (such as Muslim holy book) as "مُحَمَّد" whereas it is written as "محمد" without any short vowel or diacritic in almost all other documents. The diacritics appear on the Arabic proper noun "محمد" are: 'a' = 'ً', 'o' = 'ُ' and a repeated consonant 'm' that is considered as a diacritic 'ّ'. The correct transliteration of "مُحَمَّد" is "Mohammad", where the Romanization of "محمد" is "mhmd". If we use the Dice's approach to measure the similarity between "Mohammad" and "mhmd", the score will be 0 (SIM("mohammad", "mhmd") = 2*0/(3+7) = 0). Hence many correct transliteration pairs have low Dice's Similarity Coefficient value that forces the system to discard them.

3.4 Experimental results using SIM1

As illustrated in the previous section, the Dice's approach to measure similarity between English –

Arabic transliteration pairs does not work well. It leads us to use another approach of similarity called "SIM1" as illustrated in section 2.2.

Apply the algorithm of section 2.2 on the transliteration pair "mhmd = محمد, mohammed", SIM1 = 4/8 = 0.5 instead of 0 in the case of using Dice's approach. Hence, if the threshold "th" = 0.5, the transliteration pair "mhmd = محمد, mohammed" will be included in the final file. Table 2 show the results when we apply the algorithm in section 2.2 to specify SIM1 as a similarity score for the transliteration pair under consideration.

Table 2: The results using SIM1 Similarity Coefficient

th	1.0	0.9	0.8	0.7
Precision	100%	100%	92.4%	75.7%
Recall	2.3%	6.5%	26.7%	45.2%
F-Measure	4.5%	12.2%	41.4%	56.6%
th	0.6	0.5	0.3	0.0
Precision	61.8%	36.3%	22.1%	24.2%
Recall	57.1%	62.4%	78.7%	98.1%
F-Measure	59.4%	45.9%	34.5%	38.8%

It is clear from table 2 that the recall has been improved compared with table 1. However, the precision is slightly decreased. For th = 1, all the Romanized Arabic characters are matched such as "alahram, الأهرام" (the Romanization form of "الأهرام" is "alahram"), "mark, مارك" and "taba, طابا".
For th = 0.9, almost one character in a long word is not mapped properly, such as "kazakhistan, كازاخستان = kazakhstan", only the short vowel 'i' does not appear in the Arabic word. The same word "kazakhistan" appears in th = 1 as "kazakhstan, كازاخستان".

For th = 0.8, most of the error occurred with short words that has matching score equal or more than 0.8. For instance the transliteration pair, "aladl, الامل = alaml", however, 4 characters are matched this transliteration is not correct. It is better to match all consonant and long vowel characters of the converted word to avoid this kind of error. Other example: "alalam, الإسلام = alaslam". Another problem is: "salam, سالم = salm", however all the converted word characters are matched, the transliterated pair is not correct. Some others have more than one transliteration and all are correct such as "tahir, طاهر = tahr, taher". This occurs since the Arabic language has only three short ('a', 'e', 'o') and three long ('a', 'y', 'w') vowels. Hence the language does not differentiate between the English vowels 'i' and 'e'. Another example occurred because of different pronunciation of the people such as: "alsobah, الصباح = alsbah, alsabah". For th = 0.5, the correct pairs are few such as "Mohammad, محمد".

3.5 Experimental results using SIM2

As we notice in the previous section, in the transliteration pair "aladl, الامل", when the Arabic word "الامل" is converted to English alphabet, it will be "alaml". If we match "aladl" with "alaml", only 'd' and 'm' do not match. So the similarity score SIM1 = 0.8. And the pair is not correct. Hence, it is required that the system restricts the extracted transliteration pairs only to the pairs that have all Romanized Arabic characters matched with some or all English proper noun characters to increase the precision. We achieve that by modifying the algorithm mentioned in section 2.2 to set the similarity score to zero if any Romanized Arabic character does not match with any English character. Hence we use a new similarity measure called SIM2. SIM2 = 0 if any Romanized Arabic character does not match with any English character. Using SIM2 as a similarity measure, we achieved the results in table 3.

The drawback of this restriction is that the recall is slightly decreased. Although the precision is increased in general, there are some errors. For instance: "hamdi, حامد = hamd". However all Romanized Arabic characters are matched with some English characters, the transliteration pair is not correct.

Table 3: The results using SIM2 Similarity Coefficient

th	1.0	0.9	0.8	0.7
Precision	100%	100%	98.9%	94.5%
Recall	2.3%	6.5%	24.6%	42.3%
F-Measure	4.5%	12.2%	39.4%	58.4%
th	0.6	0.5	0.3	0.0
Precision	88.6%	56.1%	34.2%	24.2%
Recall	53.4%	60.3%	66.4%	98.1%
F-Measure	66.6%	58.1%	45.1%	38.8%

4 Conclusions and future work

In this study, we presented a new system to create English- Arabic transliteration pairs from parallel corpora based on different similarity measure

approaches. The strength of our new system is that it works well for low-frequency transliteration pairs. The system could extract some correct transliteration pairs of frequency equal to 1. We found that the similarity measure must be specified based on the characteristics of the two languages pair under consideration. We have evaluated the presented new approaches using the English- Arabic parallel corpora. Most of our results outperform previously published results in terms of precision, recall and F-Measure. We believe that the presented approach will improve the precision and recall in cross language information retrieval system.

In the future work, we will use the resulted transliteration pairs in cross language information retrieval and machine translation systems.

Acknowledgment

Authors are grateful acknowledge support from the Japan Society for the Promotion of Science (JSPS), Grant No. 07077.

References:

[1] Al-Onaizan, Y., Knight, K. (2002). Translating named entities using monolingual and bilingual resources. *In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), Philadelphia, 2002, 400–408.*

[2] Stalls, B., Knight, K. (1998). Translating Names and Technical Terms in Arabic Text. *Proc of the COLING/ACL Workshop on Computational Approaches to Semitic Languages.*

[3] Chen, H. H., Huang, S. J., Ding, Y. W., Tsai, S. C. (1998). Proper name translation in cross-language information retrieval. *In Proceedings of 17th COLING and 36th ACL, 1998, 232–236.*

[4] Knight, K., Graehl, J. (1998). Machine transliteration. *Computational Linguistics 24 (4), 599– 612.*

[5] Kang, B. J., Choi, K. S. (2001). Two approaches for the resolution of word mismatch problem caused by English words and foreign words in Korean information retrieval. *International Journal of Computer Processing of Oriental Languages 14 (2), 109–131.*

[6] Fattah, M., Ren, F., Kuroiwa, S. (2006a). Stemming to Improve Translation Lexicon Creation form Bitexts. *Information Processing & Management. 42, (4), 1003-1016.*

[7] Fung, P. (1995). A Pattern Matching Method for Finding Noun and Proper Noun Translations from Noisy Parallel Corpora. *CoRR cmp-lg/9505016.*

[8] Fung, P., Yee, L. (1998). An IR Approach for Translating New Words from Nonparallel, Comparable Texts. *COLING-ACL 1998: 414-420*

[9] Fung, P. (1998). A Statistical View on Bilingual Lexicon Extraction: From Parallel Corpora to Non-parallel Corpora. *AMTA 1998: 1-17*

[10] McEnery, A. M., Oakes, M. P. (1996). Sentence and Word Alignment in the Crater Project. *In J. Thomas & M. Short (eds.), Using Corpora for Language Research, London: Longman, 211-231.*