

Creating Lexicon From Bitexts and Effect of Stemming

Presented By: Hassan S. Al-Ayesh

Outlines

- Introduction.
- Extracting Parallel Documents from the Internet.
- Preprocessing.
- The System.
- The First Algorithm.
 - Examples.
 - Results.
- The Second Algorithm.
 - Examples.
 - Results.
- Effect of Stemming.
- Conclusions.
- Reference.

Introduction

- Stemming: Process of normalizing word variations by removing prefixes & suffixes.
- English-Arabic Parallel documents can be used to build Arabic-English Dictionary, However they are hard to find.
- Main Providers for these Documents are the newspapers, magazines & News websites.
- Bitexts or Parallel Corpora are bodies of texts in Parallel translation.
- In Following slides, A system will be demonstrated for creating English-Arabic Bitexts.

Extracting Parallel Documents from the Internet

- Steps Required to Find Parallel Documents:
 1. The pages that might contain parallel documents located using some search queries like “Arabic Version”, “English Version” and so on.
 2. Download or Generate the Documents Pairs.
 3. Filter out the non-translation candidates Pairs.
- The Types of Documents Collected are:
 - a) Parent Page: Is the one that contains links to different languages versions.
 - b) Sibling Page: Is a page in one language that contains a link to another language version of the same page.

Preprocessing

- Preprocessing involves:

1. Align the sentence pairs based on their length.
2. Remove the English and Arabic stop word lists from these documents.
 - English: possessive pronouns, pronouns, prepositions and some words that has no candidates like a, an and so on.
 - Arabic: pronouns, prepositions and some words like و، أن، إن، لقد and so on.
3. Delete some symbols and remove diacritics from Arabic texts.
4. Convert plural words to singular.

The System

- The System contains of:
 1. The Searcher.
 2. The Preprocessor.
 3. The Stemmer with the Two Algorithms that will be Discussed Later.

First Algorithm

- The similarity metric between English and Arabic words based on statistical co-occurrence and frequency of English & Arabic words.
- First make a table that contains the word, sentence numbers in which the word occurred & the frequency of the word.
- Then use the Algorithm in Next page.

First Algorithm (continued)

Set $i=j=1$

Test:if $(m_{i,j} \geq x \cdot a_{ni}) \&\& (y \cdot e_{nj} = z \cdot a_{ni})$

 CopyArabicWord(i)&CopyEnglishWord(j) to Final Document

End if

$j = j+1$

If $(j \leq NE)$

 Goto Test

End if

$i = i+1 \& j = 1$

If $(i = NA)$

 Goto Test

End if

Where $(m_{i,j})$ is number of occurrence of Arabic and English word in same sentences, (a_{ni}) is the frequency of Arabic word, (e_{nj}) is the frequency of English word, (i) Arabic word selected, (j) English word selected, (x,y,z) system parameters, (NE) is total number of English words and (NA) is total number of Arabic words.

First Algorithm (Example)

Assume the following English sentences:

- (1) Swimming is a popular sport.
- (2) Basketball was considered as the popular game in USA.

The Arabic translations are:

- (1) السباحة رياضة محببة.
- (2) كرة السلة تعتبر لعبة محببة في الولايات المتحدة.

First Algorithm (Results)

- To find the results we calculate:
 1. Precision = Correct / (Correct + Wrong)
 2. Recall = Correct / (Correct + Missing)
 3. $F = (2 * \text{Precision} * \text{Recall}) / (\text{Precision} + \text{Recall})$
- The effect of (M_i, j) on the precision, recall and F -measure:

M_i, j	$M_i, j > 2$	$M_i, j > 4$	$M_i, j > 6$	$M_i, j > 8$	$M_i, j > 10$	$M_i, j > 12$
Precision	44.3%	75.7%	82.5%	86.3%	95.6%	100%
Recall	33.8%	11.8%	6.2%	5.4%	3.9%	2.3%
F -measure	38.3%	20.4%	11.5%	10.2%	7.5%	4.5%

First Algorithm (Results) Cont.

- (mi, j) is directly proportional to the precision of the resulted dictionary, and inversely proportional to the recall.
- Advantage: The Algorithm is efficient for big corpora.
- Disadvantage: Fail to capture dependencies between group of words.

Second Algorithm

- Based on statistical co-occurrence of pairs.

FirstStep:

Set $n=1$, & $m=n+1$.

Test: Compare Arabic_sentence (n) with Arabic_sentence (m)

Compare English_sentence (n) with English_sentence (m)

If only one Arabic word common between Arabic_sentence (n) and Arabic_sentence (m)

Copy the Arabic word and the associated English word or phrase in a table

End

$m=m+1$

If $m \leq N$

GOTO Test

End If

$n=n+1$ & $m=n+1$

If $m \leq N$

GOTO Test

End If

Then Exchange Arabic by English, and then repeat the previous pseudo code.

Where N is number of sentences.

Second Algorithm Cont.

- The output of previous Algorithm is:
 - a- One English word translated to one Arabic word.
 - b- One English word translated to an Arabic phrase.
 - c- One Arabic word translated to an English phrase.
- Then use that Algorithm:

For i=1; i <= Na_word; i++

Get all English_words associated with Arabic_word (i)

For all English_words associated with Arabic_word (i)

If $R(e) > Th$

Copy Arabic_word (i) & "e" in a final file

End IF

End For

End For

Where Na_Word is the total number of Arabic words in the table, $R(e) = f(e)/NE$ is The Repetition percentage of English word e and f(e) is the frequency of English word e .

Second Algorithm (Example)

Assume the following English sentences:

1. I can play football.
2. Football is a popular sport.
3. Basketball was considered as the popular game in USA.

The Arabic translations are:

١. استطيع ان العب كرة القدم.
٢. كرة القدم رياضة محببة.
٣. كرة السلة تعتبر لعبة محببة في الولايات المتحدة.

Second Algorithm (Results)

- The precision and recall of translation pairs resulted from applying the previous algorithm depend on the value (Th) in which the precision is directly proportional to (Th), and the recall is inversely proportional to (Th).
- The effect of (Th) on the precision, recall and F -measure:

h	0.20	0.40	0.60	0.80	0.99
Precision	69.2%	76.0%	83.8%	85.0%	85.3%
Recall	90.0%	84.6%	75.9%	75.0%	74.5%
F -measure	78.2%	80.1%	79.7%	79.7%	79.5%

Second Algorithm (Results) Cont.

- The effect of trial number on the precision, recall and *F*-measure:

Trial	First	Second	Third	Fourth	Fifth	Sixth	Seventh	Eighth
Precision	86.0%	91.4%	85.3%	81.6%	76.7%	73.1%	74.3%	68.7%
Recall	8.0%	45.8%	15.4%	6.1%	2.3%	1.4%	1.0%	0.7%
<i>F</i> -measure	14.6%	61.0%	26.1%	11.4%	4.5%	2.7%	2.0%	1.4%

- Disadvantage: The processing time required for algorithm 2 is higher than that of algorithm 1.
- Because of disadvantages of algorithm 1 & 2 it is better to use combination of these Algorithms.

Effect of Stemming

- Using “Aitao Chan & Ferdric Gey” Arabic Light Stemmer.
- That Stemmer removes Prefixes and Suffixes in that sequence:
 1. If the word is at least five-character long, remove the first three characters if they are one of the following: بال، فال، كال، ولل، مال، ال، سال، لال، وال.
 2. If the word is at least four-character long, remove the first two characters if they are one of the following: با، لل، وم، وت، وب، لا، وب، سي، وس، وي، واء، ال، فا، كا، ول، وي، وس، سي، لا، وب، وت، وم، لل، با.
 3. If the word is at least four-character long and begins with و, remove the initial letter و.
 4. If the word is at least four-character long and begins with either ب or ل, remove ب or ل only if, after removing the initial character, the resultant word is present in the Arabic document collection.
 5. Recursively strips the following two-character suffixes in the order of presentation if the word is at least four characters long before removing a suffix: ون، ات، ان، ين، تن، تم، كن، كم، هن، يا، ني، وا، ما، نا، هم، ية، ها.
 6. Recursively strips the following one-character suffixes in the order of presentation if the word is at least three-character long before removing a suffix: ت، ي، ه، ة.

Effect of Stemming

- The system accuracy increased but the total recall decreased.
- The accuracy increased because of the decrease of the system confusion due to the increase of the translation pair frequency after stemming.
- The recall decreased due to that many Arabic words have been reduced to one word after stemming.
- The accuracy did not increase too much after stemming because the formation of broken Arabic plurals is complex and often irregular. Like ادوات after stemming becomes ادو which is not right word.

The Output

- A part of the final output file:

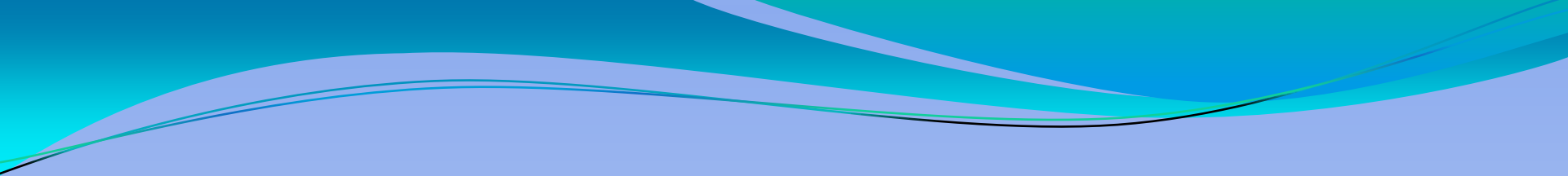
غفور	Oft-forgiving	1	يا	O	1
اتفاق	Agreement	1	ليل	Night	1
اجتماع	Meeting	1	إلا	Except	1
اختيار	Choice	1	أو	Little	0
استخدام	Fragmentation	0	قليلا	little	1
استعراض	Review	1	رب	Lord	1
استكشاف	Exploration	1	إله	God	1
استنادا	Based	1	يقول	Say	1
اقتصاد	Economy	1	أرض	Earth	1
اتحاد	Union	1	يوم	Day	1
صم	Deaf	1	جبال	Mountain	0
اتصال	Communication	1	فصل	Day of sorting out	1
اتفاق	Agreement	1	قارعة	Day of noise and clamour	1

Conclusion

- The first algorithm achieved high precision with low recall for high frequency words and its required processing time is small. However it failed to handle compound nouns
- Algorithm 2 can handle the translation of compound nouns with high precision and recall, but it needs long time.
- Stemming as a preprocessing step has increased the system accuracy but it has decreased recall.

Reference

- Stemming to Improve Translation Lexicon Creation from bitexts by Mohamed Abdel Fattah, Fuji Ren and Shingo Kuroiwa.



Thank You...
If you have any Question, Just Ask.