A Hybrid Model for Autonomous Danish-Arabic Statistical Machine Translation

Mossab Al-Hunaity

Abstract: ***We present a simple and efficient method for enhancing the Danish-Arabic (DA-AR) statistical machine translation system. The model mainly is composed of two major parts, information retrieval unit and SMT system. We train our baseline with small DA-AR corpora. We use the Arabic translation output as a query to Lemur information retrieval tool to search for a similar matching sentence in a very larger Arabic corpus. We use Translation Error Rate (TER) filter to select the best output of the IR system. We evaluate our approach and prove that it enhances the quality of translation. We extend our experiments to measure the effect of adding more language resources to our baseline. We mine available DA-EN and EN-AR resources to produce parallel DA-AR sentences. We use the new resources in training our baseline. We evaluate the quality of the extracted data by showing that it significantly improves the performance of our baseline performance.***

Keywords: (DA-AR), (TER), Danish-Arabic , DA-EN and EN AR, baseline performance

I. INTRODUCTION

Developing a statistical machine translation (SMT) system for languages with limited common bilingual resources like the case with Arabic (AR) and Danish (DA) is a challenging task. To produce good results SMT system needs huge amounts of bilingual data to produce reasonable translations. The challenging question is how our SMT system can produce a reasonable translation without having enough bilingual training data. To answer this question we started from the observation that, for a domain specific text, like the computer printer manuals, usually sentences are repeated through many different printer manuals. Sentences will have similar structure and it's very common to find similar sentences between many printer manuals with the same meaning and structure, like for example: "turn on your printer" or "insert installation desk in computer drive", etc. Now if we have an SMT system that is trained on small amounts of language resources then the output translated sentence would have a weak syntactic and grammatical structure, but still normally it has the meaning. So we use this fact to correct our SMT translation. For example, if we were to translate between two languages with limited resources like Swahili and English where Swahili is the source language and English is the target, and let's assume that we are trying to translate a computer printer manual. It is very likely to produce unreasonable translation so we might have an output translation like "printer insert disk", but if we search our English manual we will find that the most suitable sentence that is composed of similar numbers of Swahili words and have "printer insert disk" would be "Insert printer installation disk into drive".

Manuscript Received on June 2014.

Mossab AL Hunaity, Department of Computer Information Systems, Ajman University, Ajma, United Arab Emirates. E-mail: mossab99@gmail.com Mobile: +962777496676

So we search for the possible and correct translation in the target language corpus to find the correct translation. We argue that if you have a very large domain specific corpus for a target language, then it is very likely that you will have a similar sentence that matches the meaning of your baseline translation. We will apply the same idea on our DA-AR case. So we build a baseline that is trained on a small bilingual corpus and a DA-AR dictionary where Danish is the source language and Arabic is the target language. We utilize the Lemur information retrieval tool to search for all possible similar sentences in a very large Arabic corpus that has that is domain specific to the baseline corpus. We managed to find correct translations similar to our baseline translated output. This approach enabled us to overpass the obstacle of having limited resources between Danish and Arabic. Although the system may not produce literal translation in many cases, it still produces a comparable translation that is syntactically and grammatically correct and still conveys the same meaning as the original sentence. And we think this is how human translation tends to be, it is more "meaning oriented" rather than "literal oriented" process.

We also try to enhance our baseline performance by mining all possible parallel DA-AR text from DA-EN, and EN-AR resources that are freely and widely available. We retrain our baseline with the extracted data. We report performance improvements on our system BLEU scores. In the next section we describe related work. Section 3 presents our system description. In section 4 we describe our data resources. We present our system architecture in section 5. In section 6 we present and explain the results of our experiments. Finally we discuss our conclusions and future work in section 7.

II. RELATED WORK

The idea of using information retrieval techniques to enhance SMT systems was introduced by many researchers, for example, Oard (1997), Och et.al (2002,2003). Eck et al. (2005) they proposed a new technique to select matched sentences based on n-gram coverage, n-grams were used to figure out how important the sentence was. The selected the frequency of the n-gram appearance is regarded as an indicator for selection, unseen n-grams were favored for selection. TF-IDF weighting scheme was tested in this method but no improvements was reported over n-grams, the idea was to decrease the amount of training data so that to make it possible to be deployed for small devices like PDA. Hildebrand et al. (2005) used information retrieval method for translation matching, they start with an adaptable translation model and they select similar sentences from a test set from in-domain and out-of domain training data.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

We use the same approach almost with a concentration on "in-domain" training data to detect a candidate translation in the in-domain text. Zhao et al. (2004) and Eck et al.(2004) they developed many experiments to use information retrieval as an adaptation model for supporting their SMT systems. Zhang et al (2006) and Mauser et al. (2006) use a new modified language model for their SMT, where one translation is used for re-ranking the translation output , the language model is built for the target language for the SMT , n-best translation candidate is generated after the first pass for the baseline SMT. In our work we try to boost also system performance by extraditing parallel Danish-Arabic sentences from available resources. Many researches inspected the problem of enhancing the performance of SMT systems by discovering parallel sentences from different linguistic resources. Resnik and Smith (2003) propose their STRAND web-mining based system for mining parallel sentences. The system manages to find large number of similar documents. Utiyama and Isahara, (2003) also developed another interesting technique for finding parallel sentences from comparable corpus between Japanese and English using dynamic programming and cross language information retrieval methods. Their approach was to identify similar article pairs and then treat these pairs as parallel text, then they tried to find similar sentences with similarity score and document pair measures , they would declare a match based on the least-cost alignment over the document pair measure. We use a similar approach but for finding a similar translation based to our baseline output. Yang and Lee (2003) they use dynamic programming to find parallel sentences in title pairs, they calculate confidence score to find a match between sentences which is based on longest common subsequence, In our work we use a similar approach for finding a matching sentence pair. Fung and Cheung (2004) used the same IR approach, they deployed the cosine similarity to match documents, they worked on noisy comparable corpora, their method will generate many candidate parallel selections where the best match is selected based on a threshold of cosine similarity scores, the use the extracted sentences would serve to build a dictionary to enhance their SMT system. Munteanu and Marcu (2005) uses bilingual dictionary to translate some of the words of source sentences. These translations are then used to query another system to find a matching translation using IR techniques. Candidate sentences will be selected based on words overlap. The maximum entropy classifier is used to select the best matching parallel sentences. Increasing the size of the bilingual dictionary and the Bootstrapping method is used to produce better results. Our technique is similar to that of Munteanu and Marcu (2005) but we don't use a dictionary for sentence discovery, instead we use our baseline output translation as a reference for our sentence matching query. Abdul-Rauf , Schwenk (2009) suggested a similar approach to Munteanu and Marcu, (2005), They extract parallel sentences from comparable corpora using IR techniques and then use that as an input for their SMT baseline , the translation is used again with another IR tool to find parallel sentences. Our work is differs from their approach in that we don't target parallel sentences rather than we search for similar comparable sentences as a target translation.

III. BASELINE DESCRIPTION

Our system is based on the Moses SMT toolkit (Koehn et al., 2007), we intend to translate from Danish to Arabic. The system is constructed as follows. First, we use Giza++ to perform word alignments in both directions. Second we extract phrases and lexical re-orderings using the default settings of the Moses SMT toolkit. The 4-gram back-off target LM is trained on the 3.0 Giga words Arabic monolingual Gigaword corpus. The Danish text we intend to translate to Arabic shares the same nature as the Arabic monolingual corpus therefore, it is likely that the target Arabic language model includes at least some of the translations of the Danish text , section 4 explains experiment data in more details . We argue that this is a key factor to obtain good quality translations. The translation model was trained on the Arabic Novels translated to Danish corpus (1.3 M words) and a DA-AR dictionary of about 500k entries. In a different version of this system, more language resources like Europarl [1] (47M words) , Acquis [2] corpus (31M words), United Nations [3] corpus, Meedan [4] translation memory and LDC [5] (catalog no. LDC2004T17) were added later to our experiments to boost baseline performance. The system interacts with Lemur information retrieval system, see section 5, and use the output of the text mining engine that we develop to increase our DA-AR parallel resources.

IV. DATA

In our experiment we use two sources of data; parallel DA-AR resources and comparable DA-EN, EN-AR language resources. For our baseline we need a common bilingual DA-AR data source. Unfortunately parallel DA-AR resources are not very common, but we managed to develop a 1.3 M parallel DA-AR corpus. It is based on Arabic novels that were translated for Danish. We divide this corpus into two sets; training set (1M) and testing set (300 K). These novels were written in Modern Standard Arabic. Data were processed for typing mistakes and were tokenized before being used. We also prepared a 1M word monolingual Danish corpus that is similar in text to the training data for testing purposes as well.

For our language model, we collected a 3.0 Giga word monolingual Arabic corpus that shares the same text (Novels) with the training data. Data was processed for typing mistakes and text was tokenized the same way we did with training data. Table 1 explains our experiments data sources. It is important to notice that the Arabic language model includes somehow the translation of the Danish sample test data. We argue that this is a key factor in our experiment to have a quality of translation.

The baseline will produce a translation for the input Danish sentence. This Arabic translation might contain problems like grammatical or syntactical errors or word ordering problems (Verb, SUB, OBJ). We solve this shortcoming by searching for a similar sentence in the large Arabic corpus. Later we output the most relevant Arabic sentence to the baseline translated Danish sentence to be a possible translation for the system instead of the baseline original translation. The benefit of this approach is that we will utilize the Arabic LM corpus to search for a suitable translation.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

We are aware of the fact that novels as a text are not the best source of data you can get for your SMT system. A better source might be as the one presented by our introduction example, a domain specific text like hardware manuals, climate and whether change reports, etc. but in our case where only limited resources are available between Danish and Arabic we need to invest in these available resources. We selected our test data carefully to make sure that the content of the test data is contained in the large Arabic monolingual corpus so that we can test our theory. A major goal we are trying to achieve here is how we can extract knowledge (translation) out of noisy environment. We discuss our approach in details in section 5.

Another factor we try to inspect in our experiment is the additionalthe additional out of domain mined data in our baseline performance, for this purpose we will make use of the available comparable language resources for the two pairs of languages DA-EN and EN-AR. For the DA-EN group we will be using Acquis and Europarl parallel corpora. Acquis is collection of legislative texts from the European Union (EU) member states parliament meetings, while the Europarl parallel corpus is extracted from the proceedings of the European Parliament. Both Acquis and Europarl data domain is formal and covers the legal issues and they are free to use. For the Arabic-English pair we selected three major resources, the United Nations (UN) multilingual corpus, Meedan translation memory and LDC (catalog no. LDC2004T17). Both UN and Meedan are free to use and download. The domain of this group is mainly news, table 1 explains more details about our data resources.

1: http://www.statmt.org/europarl/
2: Acquis http://langtech.jrc.it/JRC-Acquis.html
3: UN Corpus http://www.uncorpora.org/
4:Meedan http://github.com/anastaw/Meedan-Memory
5: LDC http://www.ldc.upenn.edu/

Both groups (DA-EN, EN-AR) inter cross partially in domain, so we can classify the two groups as a partially comparable resources which will be used to extract parallel Danish-Arabic sentences. Later this data will be used to boost our baseline performance. Section 5.3 will provide more details about this process details.

Table 1: Data Resources

Name	Direction	Domain	Size (words)
Training Corpus	Danish-Arabic	Arabic Novels	1.3 M
LM Corpus	Arabic	Arabic Novels	3.0 G
Europarl	Danish-English	Legal	47 M
Acquis	Danish-English	Legal	31 M
UN multilingual corpus	Arabic-English	Legal issues / News	3.2 M
Meedan	Arabic-English	News	0.5 M
LDC2004T17	Arabic-English	News	0.5 M

V. SYSTEM ARCHITECTURE

The general architecture of our system is shown in figure 1. Starting from a small parallel AR-DA corpus and an AR-DA dictionary we build a basic baseline see Section 3 for details.. To ensure a correct and accurate translation we use system output translation which may have syntactical or grammatical problems to search a very large corpus for similar sentences. We use Lemur information retrieval system to extract the most similar sentence to our baseline translation output.

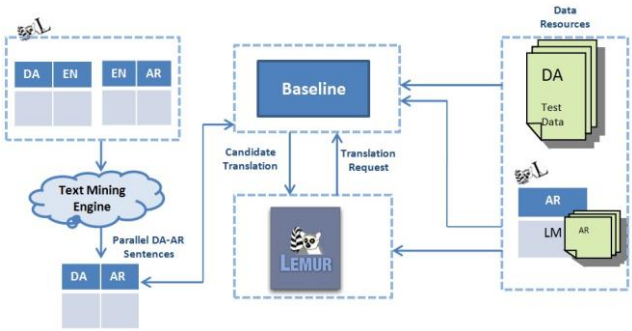

Figure 1: System Major Architecture

We check the quality of our selected sentence using simple metrics like Word Error Rate (WER) and Translation Error Rate (TER) which will help us filter out good sentence matching pairs. If we find a good matching sentence then we replace the original baseline sentence translation with the new found Arabic sentence. Eventually we store the original Danish sentence and its translation into our DA-AR parallel corpus. We retrain the baseline again with the new bilingual data. We show that a parallel corpus obtained using this technique helps considerably to improve our SMT baseline. Sections 5.1 and 5.2 describe this process in details. We inspect another approach to improve our baseline performance. We mine available DA-EN and EN-AR language resources to extract parallel DA-AR sentences. We try to find all common English sentences between the two groups. Figure 1 explains this process. Once we have a match we extract the Danish sentence from the DA-EN part and the Arabic sentence from the EN-AR part. We store the new parallel extracted sentence pair into our DA-AR corpus. This step will help increase our parallel Danish-Arabic data size. To achieve that first we use Lemur to build indexes on our EN-AR resources. We start the search from the DA-EN side. We prepare a window size of four words and search all the possible occurrences of that window in the EN-AR side. If we find a match then we extract the Danish part that is parallel with the English window. We do the same with the match English sentence and extract the parallel Arabic part of that sentence. Lemur IR system is used to minimize the search space while searching for matches and will locate the most likely document that contains a matching English sentence at the EN-AR side. We apply the longest common subsequence problem approach to find a similar common string between resources. Section 5.3 explains more details of our approach. We retrain our baseline with the new extracted data, this will help boost the baseline performance. Finally we show that new parallel data extracted from that step which is not similar to the baseline training data domain will help improve the system performance, but not significantly.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

Table 2: Translation examples produced by our baseline and query results returned by Lemur IR tool

Example 1	
Danish Source	Hun svarede mig ikke og jeg fortrød med det samme, at jeg havde spurgt.
English Translation	She did not answer my question and I immediately regretted having asked
Arabic Reference	لم تجبني .. وشعرت بالندم مباشرة بعد ما سألتها
Query (System Translation)	لم تجبني .. وشعر ندم فورا بعد سؤال
Result 1	*لم تجبني* , أحسست بالندم مباشرة بعدها على سؤالي
Result 2	*لم تجبني* ,اقتربت اكثر فاكثر، ولم*تجبني* ،مددت يدي وامسكت بكتفها *وسألتها*
Result 3	*لم تجبني* فقد نهضت وهي تنظر بندم لأحد الطاولات خلفنا، *بعد ما* ألقاه من كلام*لم* استطع النطق مباشرة ولا حتى التعليل

Example 2	
Danish Source	Jeg er sikker på, at vi snart får dem at se igen," sagde jeg, i et forsøg på at glatte ud.
English Translation	I am sure we will see them again. "I said that in an attempt to smooth things over"
Arabic Reference	انا متاكد اننا سنراهم مرة اخرى, لقد قلت ذلك لتلطيف الامور
Query	انا متاكد سنراهم رأي قال ذلك تحسين الاشياء
Result 1	انا متاكدة اني سارى الامل مرة اخرى , قلت ذلك لتحسين الامور بيني و بينه
Result 2	لكن الشيء الذي *أنا متأكد* منه أن طالبان ستصمد. *واننا سنراهم* مرة اخرى, هذا راي
Result 3	*أنـــا متأكد* بأني قادر على كسب الثقه و *أنا* لست خائف من المنافسه . *وانهم سيغيرون رائيهم* بسرعه في علاء

Example 3	
Danish Source	Fra da af begyndte mine tanker at bevæge sig i helt nye retninger, hvoraf den første var en dyb sorg.
English Translation	From that day, my thoughts went to new directions, instead of my old deep sorrow.
Arabic Reference	ومنذ ذلك اليوم اخذت افكاري منحا جديدا , بدلا عن احزاني القديمة
Query	من اليوم ذهبت افكار جديدة بدل عميق احزان
Result 1	اصبحت افكرباسلوب جديد , لم اعد ذلك الانسان السابق , ان احزاني لن تثنيني عن المسير
Result 2	الافكار الجديدة اليوم هي العميقة بالمشاعر و الاحزان
Result 3	من الان ابدؤوا في زرع *افكار* السلام الجديدة و السعادة و الرضا , فعندما يحدث ذلك عليكم ان تتكلموا بحزم و احساس عميق بالمسؤالية

Table 2 shows an example of our best system results. The query presented in Table 2 represents the baseline translation and the results are the sentences extracted from Lemur. We aim to replace inaccurate translation with most similar sentence extracted from our Arabic monolingual corpus; extracted sentences can convey the same meaning and enjoy a correct syntactical and grammatical structure.

VI. SYSTEM FOR EXTRACTING CORRECT ARABIC TRANSLATION FROM LARGE ARABIC MONOLINGUAL CORPUS

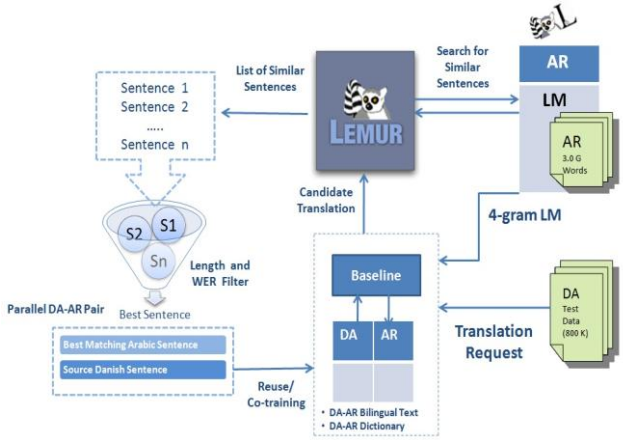

Figure 2: System for Extracting Correct Translation

The general architecture for our correct sentence translation extraction system is show in figure 2. Our baseline which is explained is section 3 will translate from Danish into Arabic. It's trained on a small bilingual dataset of 1 M words and a DA-AR dictionary of 500 K entries. For our testing data we use a sample of 300 K words of bilingual DA-AR parallel text. We use a huge monolingual corpus of 3.0 G words. All data used in this experiment shares the same text nature of Arabic novels written in modern standard Arabic. Our baseline is trained on Moses package. We use the Arabic monolingual corpus mentioned in section 4 to build our language model, we use SRILM toolkit Stolcke (2002) for that. The baseline system receives a translation request and will produce a possible translation. We notice that the output of the translation is not accurate in some cases but that is expected due to the small training resources for our baseline. Problems with translated output like grammatical and syntactical mistakes may appear. So we search in the large monolingual Arabic corpus on a similar sentence for the translated output. We process small sentences, because GIZA++ usually ignores long sentences. In a similar approach Munteanu and Marcu (2005) used the dictionary to find the matching target sentence for the source sentence. We believe using a baseline has many advantages over this approach. For example Moses is phrase oriented decoder that studies the relationship between lexicons in the sentence to produce translation, so he can learn that "book" in "book a flight" refers to different translation than "library book". A dictionary will not be able to decide what the best translation for the word in a sentence is. Moses will produce the most probable translation and consequently will help guide our search to find the most suitable sentence in the Arabic corpus. We use Lemur tool kit for searching for candidate sentences similar to our baseline output sentence Ogilvie and Callan, (2001). We selected Lemur because it supports Arabic documents which are a great advantage unfortunately many modern IR tools doesn't have. Lemur will interact with Moses directly to produce a final translation.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

Moses is trained into a small training data size and that will make its ability to produce correct sentences weak. We fix this problem by using Lemur to search for the most similar sentence to Moses output translation. Both Moses and Lemur uses the large Arabic monolingual corpus to build their own language model (LM) as described in figure 2. Moses uses it's LM to detect the most probable translation for the source sentence, while Lemur build its own LM to enhance the search process on that large monolingual corpus. Both Moses and Lemur LM's are different in structure and usage, but they use the same corpus. For Lemur to function efficiently we need to build indexes on our search space. We format our Arabic monolingual corpus according to NIST [1] format, so that it can be recognizable from Lemur Arabic parser, table 3 gives an example of NIST style documents. We use Lemur Arabic language parsers to build Lemur indexes. After initial parsing we add a list of Arabic stopping words[2] to be ignored by Lemur while indexing. We use the stem indexing feature of Lemur which enhances recall results. Using this feature has shown better recall results for Lemur. We process only the top 3 scoring sentences that are returned by the IR process. We found no evidence that retrieving more than 3 top scoring sentences helped get better sentences. At the end of this step, we have for each query sentence 3 potentially matching sentences as per the IR score. The information retrieval step is the most time consuming task in the whole system. The time taken depends upon various factors like size of the index to search in, length of the query sentence etc. Query length also affected the speed of the sentence extraction process. We placed a limit of approximately 25 words on the queries and the indexed sentences. This choice was motivated by the fact that the word alignment toolkit Giza++ does not process longer sentences.

Table 3: NIST file structure

```
<?xml version="1.0" encoding="UTF-8"?>
<SRCSET setid="ALKARNAK_NAJEEB_MAHFOZ"
srclang="AR">
      <DOC docid="1" genre = "text" >

      <seg id="1.1">
                  اليوم يعتبر من الايام الحميلة , اعتقد ان اسامة لن
يرجع الى الاسكندرية قبل العاشرة.
      </seg>
      <seg id="1.2">
ممكن جدا, بالمناسبة هل سنتمكن من الذهاب الى بيت سحر , لقد اتصلت
            بي و ابلغتني انها ستكون بالبيت في المساء
      </seg>
      <seg id="1.3">
                  نعم, هذه فكرة جيدة
      </seg>
        </DOC>
</SRCSET>
```

VII. CANDIDATE SENTENCE PAIR SELECTION

When we receive the result from Lemur IR system we need to decide if the system returned sentences are parallel or not, we select sentences with the best score. We pass matched sentence pairs to the Lemur toolkit for further filters. Gale and Church (1993) alignment program is based on the assumption that longer sentences tend to be translated into longer sentences and shorter sentences tend to be translated to shorter sentences in other language.

We use the same logic for selecting our language pairs, a sentence pair is selected for further processing if the length ratio is more than 1.5, having in mind that Danish sentences are longer than their counterparts Arabic translations. We kept a relaxed factor of 1.5 as a length ratio.

Selected sentence pairs according to the previous mentioned criteria are then verified according to the WER (Levenshtein distance) and Translation Error Rate (TER). WER measures the number of operations required to transform one sentence into the other (insertions, deletions and substitutions). A zero WER means that the two sentences are identical, while lower WER means that sentence pairs are sharing most of the common words. WER can't detect a correct translation which may be different in order of the words because it work on word by word basis. We solve this shortcoming by using the TER which measures the number of edits needed to change system output into a given reference translation. TER allows words movements and thus can absorb word recording or rephrasing in translation as described by Snover et al. (2006). Generally, sentences selected based on TER filter showed better BLEU scores than their WER counter parts. So we chose TER filter as standard for our experiments with limited amounts of human translated corpus. Figure 3 shows a BLEU score comparison between WER and TER based on the size of the training data. These experiments were performed with only 1.3 M words of human-provided translations (Arabic Novels corpus).

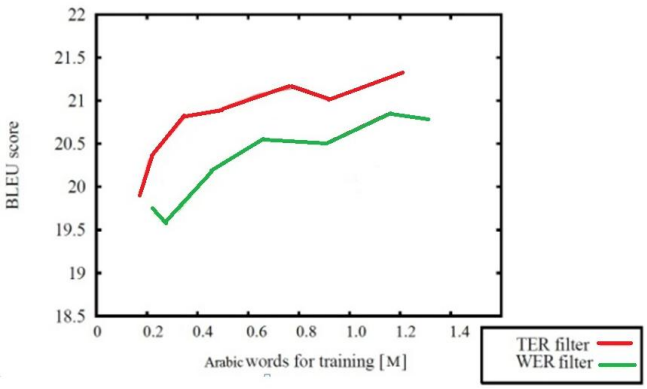

Figure 3: System performance based on TER and WER filter

1: http://trec.nist.gov/
2: http://arabicstopwords.sourceforge.net

VIII. EXTRACTING PARALLEL SENTENCES FROM COMPARABLE RESOURCES.

We intend to boost our baseline with more bilingual DA-AR training data .We extract parallel sentences form available comparable corpora. There is no direct bilingual DA-AR large corpus available, but there are many DA-EN and EN-AR corpora. We intend to use these resources to boost our baseline performance. We mine common English sentences between the two available groups (DA-EN, EN-AR). If we find a match then we extract the parallel Danish sentence from DA-EN group and we do the same for Arabic sentences with EN-AR group. We end up with a new parallel DA-AR sentences which can be used in to increase the size of our baseline data training. We start from the DA-EN group and we process every.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

English sentence as follow:

- We extract a window of 4-gram size as an input
- Lemur is used to decide what documents in the EN-AR group might contain this sentence (window). We build Lemur indexes only on the EN-AR side and in the same approach mentioned in section 5.1.
- We use the stopping word list provided by the IR Group of University of Glasgow[1] for Lemur index.
- We apply the Longest Common Substring problem on both source and target sentences, see Section 5.3.1.
- We extract the matching Danish and Arabic sentences as a new parallel pair. Table 5 provides an example of these steps.

IX. LONGEST COMMON SUBSTRING (LCS)

Finding a matching sentences between the two groups mentioned in section 5.3 is time consuming and a nontrivial task to tackle. We developed the algorithm shown in table 4 as an implementation for the LCS problem.

Table 4: LCS algorithm

```
LCS-Length(Src, Dest)

   L = length[Src]
   H = length[Dest]
   for Start = 1 to L
      c[i,0] = 0
   for j = 1 to H
      c[0,j] =0
   for i = 1 to L
          for j =1 to H
          if (x[i] == y[j]) {
            c[i,j] = c[i-1,j-1] + 1
            b[i,j] = NW
            }
          else if (c[i-1,j] >= c[i,j-1])
           {
              c[i,j] = c[i-1,j]
              b[i,j] = N
            }
          else {
            c[i,j] = c[i,j-1]
            b[i,j] = W
            }
```

The algorithm declare a two dimensional array to represent the two sentences to be matched. At the start the matrix elements are initialized with zeros as shown in table 5. Then it will search for a match on word bases, if it finds it will add 1 to the diagonal position of the cell. Finally when the Algorithm finishes all possible comparisons between words, the cell with the largest value would represent the longest sequence (sentence) out of our sentence comparisons . We extract these words (window), along with the parallel accompanied Danish and Arabic sentences. Table 4 gives an example of the algorithm major steps.

1:http://ir.dcs.gla.ac.uk/resources/linguisticutils/stopwords

Table 5: LCS Process Example

		Danish-English				
		D1	D2	D3	D4	
		E1	E2	E3	E4	
A1	E1	0	0	0	0	0
A2	E2	0	0	0	0	0
A3	E3	0	0	0	0	0
A4	E4	0	0	0	0	0
Arab-English		0	0	0	0	0

LCS array structure

			Danish-English				
			Nye	Indtil	vigtige	spørgsmål	
			Emerging	**Pending**	**important**	**Issues**	
Arab-English	العديد	**Many**	0	0	0	0	0
	العالقة	**pending**	0	0	0	0	0
	المهمة	**Important**	0	0	1	0	0
	من القضايا	**Issues**	0	0	0	2	0
			0	0	0	0	3

LCS array values after matching

			Danish-English				
			Nye	Indtil	vigtige	spørgsmål	
			Emerging	Pending	important	Issues	
Arab-English	العديد	Many	0	0	0	0	0
	العالقة	pending	0	0	0	0	0
	المهمة	Important	0	0	0	0	0
	من القضايا	Issues	0	0	0	0	0
			0	0	0	0	0

English	Pending	Important	Issues
Danish	Indtil	Vigtige	Spørgsmål
Arabic	العالقة	المهمة	القضايا

X. RESULTS AND EVALUATIONS

Our goal was to enhance our baseline performance by replacing inaccurate Arabic translation with more accurate one using large Arabic monolingual corpus. In this section we report the results of using this approach with our baseline. We conduct many experiments to measure the effect of using this process to translation quality and correctness. Table 6 shows a comparison of two baselines one developed with translation. correction approach and the other without it.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

The sentence correction approach best BLEU score was 20.63 which is +2.51 points better than the other baseline that's best BLEU score was 18.12. As expected both systems performs better when the training data increases. The correction of translation approach has shown that it is possible to produce better translation system. In the second experiment we train our baseline with the extracted text from our parallel sentences mining step discussed in section 5.3. Table 7 shows the BLEU scores of this experiment which was time consuming, but time efficiency was not a concern for us compared to the extracted sentences output. We managed to extract 500 K words from all the available resources mentioned in section 4.

Table 6: System BLEU scores

Size	BLEU Baseline	BLEU Baseline + Translation Correction
200 K	12.17	18.23
400 K	14.23	18.30
600 K	16.36	18.22
800 K	17.52	20.42
1.3 M	18.12	20.63

Our baseline system BLEU scores always increased after adding this new parallel training data. We observed that our system best score was 21.13. The addition of the dictionary data entries to the baseline performance had more positive effect (+15%) than the extracted data from comparable resources (+2%). We think this is because of the different nature of the extracted data (legal text) from the training data.

Table 7: Effect of extracted data to our system performance.

Data	Size	BLEU
Novels	0.8 M	17.41
Novels + Dictionary	1.3 M	20.63
Novels + Dictionary+ Extracted data	1.8 M	21.13

XI. CONCLUSION AND FUTURE WORK

Our motivation for this approach was to be able to improve our DA-AR SMT performance by inspecting two factors; first one is the ability to enhance the translation output by replacing the baseline output Arabic translation with a similar sentence that we retrieve from large Arabic monolingual corpus. The second factor we tackled was enhancing the baseline with parallel DA-AR text that we mine from parallel DA-EN and EN-AR resources. The lack of parallel DA-AR resources has encouraged us to think of using existing language resources available to create DA-AR parallel resources. Our experiments results indicate the validity of building a baseline with translation correction approach. Our approach mainly uses incomplete knowledge (translated sentence) to produce a complete Knowledge. Large bilingual resources are not required to produce a reasonable translation. A major focus is on the large Arabic monolingual corpus used to build Moses language model to extract correct sentences. Our system produces translations that are more "human oriented" rather than "literally oriented". Translations would carry the same meaning as source sentence but maybe with difference in sentence structure or words order but it would still hold the meaning. We are interested in using the baseline output in training other similar baseline. We plan to use more intelligent approaches in detecting similar sentence in the Arabic monolingual corpus. We will apply a more syntactical Analysis to the baseline output before searching for similar sentences like removing connected articles from the Arabic sentence and search for the word and its concepts or synonyms.

REFERENCES

1. Stolcke. SRILM– an extensible languagemodeling toolkit. 2002. In Proc. Int. Conf. on Speech and Language Processing (ICSLP), volume 2, pages 901–904, Denver
2. Almut Silja Hildebrand, Matthias Eck, Stephan Vogel, and Alex Waibel 2005. Adaptation of the Translation Model for Statistical Machine Translation based on Information Retrieval. Proceedings of EAMT 2005: 133-142.
3. Arne Mauser, Richard Zens, Evgeny Matusov, Sasa Hasan, Hermann Ney 2006. The RWTH Statistical Machine Translation System for the IWSLT 2006 Evaluation. Proceedings of International Workshop on Spoken Language Translation.:103-110
4. Bing Zhao, Matthias Eck, Stephan Vogel 2004. Language Model Adaptation for Statistical Machine Translation with structured query models. COLING-2004
5. Christopher C. Yang and Kar Wing Li. 2003. Automatic construction of English/Chinese parallel corpora. J. Am. Soc. Inf. Sci. Technol., 54(8):730–742.
6. Dragos StefanMunteanu and DanielMarcu. 2005. Improvingmachine translation performance by exploiting non-parallel corpora. Computational Linguistics, 31(4):477–504.
7. Douglas W. Oard. 1997. Alternative approaches for cross-language text retrieval. In In AAAI Symposium on Cross-Language Text and Speech Retrieval. American Association for Artificial Intelligence.
8. Franz Josef Och and Hermann Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. In ACL, pages 295–302.
9. Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. Computational Linguistics, 29(1):19–51.
10. Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In ACL.
11. Masao Utiyama and Hitoshi Isahara. 2003. Reliable measures for aligning Japanese-English news articles and sentences. In Erhard Hinrichs and Dan Roth, editors, ACL, pages 72–79.
12. Matthias Eck, Stephan Vogel, and Alex Waibel 2004. Language Model Adaptation for Statistical Machine Translation Based on Information Retrieval. Proceedings of Fourth International Conference on Language Resources and Evaluation:327-330
13. Matthias Eck, Stephan Vogel, Alex Waibel 2005. Low cost portability for statistical machine translation based on n-gram coverage. MT Summit X: 227-234
14. Pascale Fung and Percy Cheung. 2004. Mining very nonparallel corpora: Parallel sentence and lexicon extraction via bootstrapping and em. In Dekang Lin and Dekai Wu, editors, EMNLP, pages 57–63, Barcelona, Spain, July.
15. Philipp Koehn et al. 2007. Moses: Open source toolkit for statistical machine translation. In ACL, demonstrationsession.
16. Paul Ogilvie and Jamie Callan. 2001. Experiments using the Lemur toolkit. In In Proceedings of the Tenth Text Retrieval Conference (TREC-10), pages 103–108.
17. Philip Resnik and Noah A. Smith Y. 2003. The web as a parallel corpus. Computational Linguistics, 29:349–380.
18. Sadaf Abdul-Rauf , Holger Schwenk, On the use of comparable corpora to improve SMT performance, Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics, Greek 2009,16-23,
19. William A. Gale and Kenneth W. Church. 1993. A program for aligning sentences in bilingual corpora. Computational Linguistics, 19(1):75–102.
20. Ying Zhang, Almut Silja Hildebrand, Stephan Vogel 2006. Distributed Language Modeling for N-best List Re-ranking. EMNLP-2006: 216-223