Modified Dijkstra Like Search Algorithm for English to Arabic Machine Translation System

Ahmed Hatem, Amin Nassar

Elect. & Comm. Eng. Dept. Faculty of Engineering. Cairo University, Giza, Egypt
amhatem@gmail.com, aminassar@maktoob.com

Abstract. In this paper we are introducing a modified Dijkstra's shortest path algorithm used to detect the target language phrases. We list the indexes of the source sentence's words which were found in the target language corpus and create a directed graph to detect the phrases that form a shortest path walk in the graph. This method is used in a hybrid English to Arabic MT system. The system combines between rule based and example based machine translation techniques. The system uses an English/Arabic dictionary, a stemmer, search and Arabic corpus without parallel corpus. The system was examined and was found that results were promising to be used for domain specific and scarce resources translation.

Keywords: English to Arabic hybrid Machine translation. Directed Graph Decoder.

1 Introduction

Machine translation (MT) systems are divided to Corpus based Machine Translation systems (CBMT) and Rule Based Machine Translation systems (RBMT). Both types need a lot of development effort and time to create a working system. Dologlou et al., introduced a monolingual MT system (METIS-I) that can be produced with less effort. The system used a tagged and lemmatized target language (TL) corpus without the Source Language (SL) corpus. The SL corpus was replaced by a bilingual lexicon [2]. The METIS-I system was then adapted by the European consortium which were formed of the "Institute for Language and Speech Processing (ILSP) in Athens", "the Universitat Pompeu Fabra in Barcelona", "the Institute of Applied Information Sciences (IAI) in Saarbrucken" and "the Centre for Computational Linguistics (CCL) of the K.U.Leuven" to create the METIS-II project. Vandeghinste et al., described the METIS-II project as a hybrid solution to provide MT systems for the languages with little resources. The consortium built separate systems that use the Dutch, Greek, German and Spanish languages as the SL and English as the TL corpus [3].

In this paper we are describing our hybrid English to Arabic MT system that is similar to the METIS-II system described by Dirix et al., [4].

2 Arabic language challenges

Arabic language is a challenging language for MT systems development. The absence of short vowels “diacritics” that appears with the Arabic letters to disambiguate similar word forms is one of the Arabic writing challenges [5]. The Arabic language writing common mistakes that are generated from the letters’ written form on the keyboard also introduces another challenge. This challenge was solved by word normalization [6], [7].

The strong structure, high derivational nature of the Arabic language and the ability to add a large number of affixes to each word are a morphological challenge. Gender, number (single, double, plural), grammatical case and linked pronouns to the word itself are another grammatical challenge of the Arabic language [5].

The flexible order of words in the Arabic language sentence, the ambiguity of the English word in its sentence, the multiple Arabic meanings for each English word, and the availability of large number of synonyms for each Arabic word all these increase the challenge for English to Arabic MT systems.

3 System Overview

Our system is English to Arabic hybrid MT system which combines between RBMT and EBMT. It consists of a tokenizer, English to Arabic dictionary, Arabic stemmer, retrieval engine, Phrase decoder, Arabic target language corpus and its inverted indexes.

The tokenizer is used to parse the input English sentences. It takes the input sentence, parses it and generates tokens when a white space, punctuation or non alphabet character are encountered. The E/A dictionary is used to lookup the Arabic translations of the input English sentence tokens. The Arabic light stemmer is used to normalize the Arabic words returned from the E/A dictionary. It also helps to overcome the Arabic grammatical, morphological and writing challenges described in section 2.

The light stemmer is built with the same technique used by Chen et al., [5]. Figure 1 represents the input sentence data flow starting from the input sentence parsing then dictionary translation and derivatives generation using the stemmer.

A TL corpus of 29,233Arabic sentences is used. The Arabic TL corpus inverted indexes is used to save the TL corpus’s word information “sentence’s number and word’s offsets” for each Arabic word in the TL corpus. We used the same inverted indexes structure used by Manning et al., [8]. The retrieval engine is used to retrieve the Arabic words offsets and sentence’s numbers from the inverted indexes. The phrase decoder detects the phrases in TL sentence. The highest rank phrase is retrieved from the Arabic TL corpus and returned to the user.

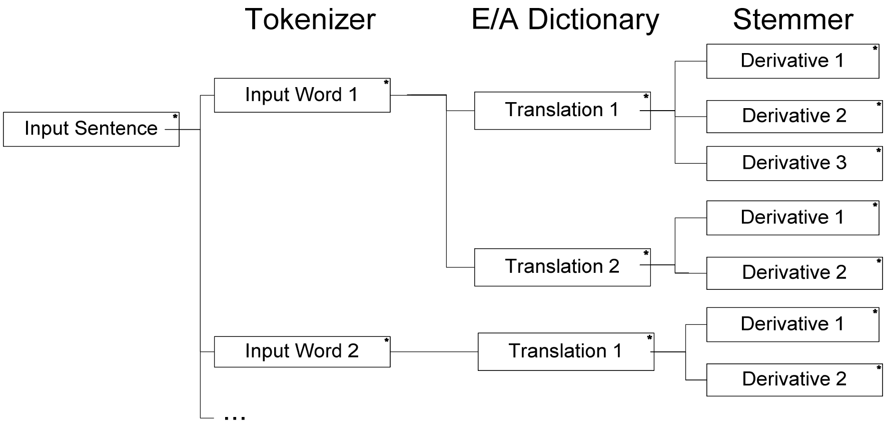

Fig. 1. Input sentence data flow

3 Phrase Decoder Directed Graph

We developed a notation to reference the SL sentence's words existence in the TL sentence. We presented the SL as a set of stream S = {s_1, s_2, …, s_n}. Where s_n represents the word s with order n in the SL sentence S. The TL output sentence T = {t_1, t_2, …, t_m}. Where t_m represents the word t with order m in the TL sentence T.

We can say that for each SL sentence S = {s_1, s_2, …, s_n}. There is a TL sentence T = {t_1, t_2, …, t_m} : t_i = {s_x, s_y, …} and $s_x \in$ S maps to the target word t_i. We get a final set T = {t_{1S3}, t_{2S2}, …, t_{mSn}}. Where t_{mSn} represent the TL word t with order m in the TL sentence T that corresponds to the SL word s with order n in the SL sentence S.

Our model is based on directed graph as in figure 2 with four dimensions (Node, Arc, Distance and Directed Walk/Phrase):

- Node: TL word corresponds to a single SL word order in the source sentence. If the TL word can map to two different SL words in the input sentence each SL word will be considered as a separate node. Nodes are represented in the form t_{mSn}.
- Arc: A connection between two adjacent word's translations in the TL sentence of two source words. It is represented as (t_{mSx}, t_{nSy})
- Arc Distance (length): The absolute difference between the source words order plus the absolute difference between target words order for two adjacent TL sentence words.

$$D(t_{mSx}, t_{nSy}) = |n-m| + |x-y| \qquad \textbf{(1)}$$

- Arc distance threshold (broken connection):

$$threshold(D(t_{mSx}, t_{nSy})) \leq 2 \ . \qquad \textbf{(2)}$$

We chose the threshold to be 2 to make sure that only target words that correspond to two adjacent source words are connected.

- Allowed Directed Walk/Phrase: Is a set of connected nodes ordered from the highest TL order to the lowest TL order. The set is either has an ascending or descending source words order. The longer the detected phrase the more natural and accurate translation is generated [9].

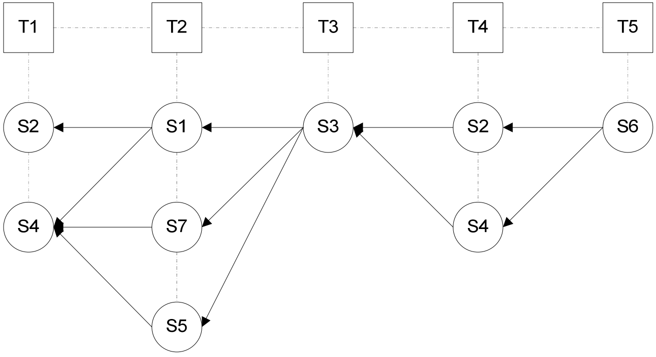

Fig. 2. Source to target directed graph

Figure 2 shows that the retrieval engine returned the target sentence T={t_1, t_2, …, t_5}. Each target sentence word maps to a certain SL word. We can find that t_1 has the same stem that maps to s_2, s_4 translations, t_1 = { s_2, s_4 }. t_2 has the same stem that maps to s_1, s_5, s_7 translations, t_2 = { s_1, s_5, s_7 }. We should then calculate the arc distance and apply the threshold to detect the adjacent phrases.

Each phrase is defined by the following attributes: phrase minimum source word order, phrase maximum source word order, phrase minimum target word order, phrase maximum target word order, phrase count of source words and phrase count of target offsets.

3.1 Modified Dijkstra's Algorithm for Phrase Detection

We developed the following algorithm to detect the adjacent phrases. This process is based on Dijkstra's [1] shortest path algorithm. However in our algorithm we don't have a start and end points to get the minimum distance between them but we traverse all the target words found. Also we don't consider the shortest path arcs only but all arcs that have distance less than the identified threshold.

We traverse the target sentence word graph from its end to start to build the phrase list $\Psi = \{P_1, P_2, \ldots P_n\}$. Where $P_i = \{t_{mSx}, t_{nSy}, \ldots\}$ is a phrase with internal distance D ≤ distance threshold between each adjacent target words of P_i. We can formalize our detection algorithm as follow: For each target sentence T that maps to the source sentence S we try to find Ψ. We use the following algorithm to get Ψ.

1. Start with an empty phrases list ($\Psi = \Phi$).
2. Traverse the target sentence graph from the last target word to the first target word. An initial phrase set $P_1 = \{t_{tailsx}: s_x \in t_{tail}\}$, $\Psi = \{P_1\}$.
3. If the tail node t_{tail} maps to more than one source word then $\Psi = \{P_1,P_2, \ldots\}$ where $P_i = \{ t_{tailSx} \}$

4. Move to the previous target word t_{i-1} and traverse all the source words $s_n \in t_{i-1}$ against all $P_i \in \Psi$ and $t_i \cap P_i \neq \Phi$. $P_i = P_i \text{ U } \{t_{i-1sn}\}$ $\forall$ $t_{i-1sn} \in t_{i-1}$ and distance $D(t_{i-1sn} , P_i) \leq$ threshold.
5. The nodes sources can either have an increasing or decreasing sequence. Otherwise the detection is branched and we have two phrases (P_i, P'). $\Psi = \Psi \cup P'$. Where $P' = \{ t_{i-1sx} , t_{isn}\}$.
6. If a broken connection is found, D > threshold. The detection algorithm add new phrase P', $\Psi = \Psi \cup P'$ and $P' = \{ t_{i-1sx} \}$
7. If all nodes are traversed break. Else go to step number 4.

The above algorithm allows detecting all adjacent words of both source sentence and target sentence with the same or reversed word mapping order.

3.2 Phrase Alignment

The same directed graph technique used by the phrase detection is used for the phrase alignment. The following rules are used to consider a connected arc between two adjacent phrases.

1. Phrase 1 max target word order < Phrase 2 min target word order.
2. Phrase 1 max source word order < Phrase 2 min source word order.
3. Phrases distance is the distance between phrase 1 and phrase 2 = |Phrase 2 min target word order - Phrase 1 max target word order| + |Phrase 2 min source word order - Phrase 1 max source word order|.
4. Phrase distance threshold is 3: If two phrases have an arc with distance > 3 then this arc will be considered a broken connection. This threshold will allow having one missing source or one extra target word.

We redraw the graph with consideration that nodes represents phrases rather than words then run the same phrase detection algorithm with the above extra constrains

4 Results

We tested our system using 100 input English sentences extracted from the United Nation English corpus. The results were categorized as Vandeghinste's [10] first experiment. We categorized our results as follow: First Rank (the number of translated sentences that got the first rank), N-Found (the number of translated sentences that were found but didn't get the first rank) and Incorrect (the number of translated sentences that were incorrect).

Table 1. Results

First Rank	60%
N-Found	8%
Incorrect	32%
Total Found	68%
Total Tested	100 sentences

As we can see from table 1 the system produced good results as 68% of the input sentences had been translated and 60% of our results were categorized as first rank.

Conclusion

Our modifications to Dijkstra's algorithm [1] can be used to detect phrases. Our hybrid MT system can be used with the Arabic language. The Arabic stemmer overcomes some of the morphological challenges that face the Arabic language translation. The TL corpus provides a context based translation guidance for the Arabic sentence. The hybrid system can be used when translating for languages with scarce resources. The hybrid system can't replace the RBMT or the SMT at this stage.

Future work

Modify the phrase detection algorithm to handle the Arabic language flexible order of words. Study the algorithm performance and complexity with bigger size corpus and larger dictionary. Use the SMT methods to detect and align the phrases. Use the semantic features of the SL and TL words.

References

1. Dijkstra, E.W.: A note on two problems in connection with graphs. Numerische Math. Vol. 1, 269--271, (1959)
2. Dologlou, Y., Markantonatou, S., Tambouratzis, G., Yannoutsou, O., Fourla, A., Ioannou, N.: Using Monolingual Corpora for Statistical Machine Translation: The METIS System. Proceedings of EAMT - CLAW 2003, Dublin, pp. 61--68, (2003)
3. Vandeghinste, V., Schuurman, I., Carl, M., Markantonatou, S., Badia, T., METIS-II: Machine Translation for Low Resource Languages. Proceedings of the 5th international conference on Language Resources and Evaluation, Genoa, Italy, pp. 24--26, (2006)
4. Dirix, P., Vandeghinste, V., Schuurman, I.: A new hybrid approach enabling MT for languages with little resources, Proceedings of the 16th Meeting of Computational Linguistics in the Netherlands, pp. 117-132, (2006)
5. Chen, A., Gey, F.: Building an arabic stemmer for information retrieval, Proceedings of the Eleventh Text REtrieval Conference (TREC 2002), NIST (2002)
6. Larkey, L.S., Ballesteros, L., Connell, M.E.: Improving Stemming for Arabic Information Retrieval: Light Stemming and Co-occurrence Analysis, In Proceedings of ACM SIGIR, pp. 269-274, (2002)
7. Attia, M.A.: Developing Robust Arabic Morphological Transducer Using Finite State Technology, the 8th Annual CLUK Research Colloquium, (2005)
8. Manning, C.D., Raghavan, P., Schutze, H., Introduction to information retrieval, Cambridge University Press, pp.3--9, (2008)
9. Doi, T., Yamamoto, H., Sumita, E.: Example-Based Machine Translation Using Efficient Sentence Retrieval Based on Edit-Distance, ACM Transactions on Asian Language Information Processing (TALIP), Volume 4, Issue 4, pp.377--399, (2005)
10. Vandeghinste, V., Dirix, P., Schuurman, I.: Example-based Translation without Parallel Corpora: First experiments on a prototype, Proceedings of the Second workshop on EBMT, pp. 135--142 (2005)