Deriving Paraphrases for Highly Inflected Languages from Comparable Documents

Kfir BAR Nachum DERSHOWITZ
School of Computer Science, Tel Aviv University, Ramat Aviv, Israel
kfirbar@post.tau.ac.il, nachumd@tau.ac.il

ABSTRACT

We describe an automatic paraphrase-inference procedure for a highly inflected language like Arabic. Paraphrases are derived from comparable documents, that is, distinct documents dealing with the same topic. A co-training approach is taken, with two classifiers, one designed to model the contexts surrounding occurrences of paraphrases, and the other trained to identify significant features of the words within paraphrases. In particular, we use morpho-syntactic features calculated for both classifiers, as is to be expected when working with highly inflected languages. We provide some experimental results for Arabic, and for the simpler English, which we find to be encouraging. Our immediate interest is to incorporate such paraphrases within an Arabic-to-English translation system.

KEYWORDS : Paraphrases, highly inflected languages, morphologically rich languages, co-training, comparable documents, Arabic

Proceedings of COLING 2012: Technical Papers, pages 185–200,
COLING 2012, Mumbai, December 2012.

1 Introduction

Paraphrases are pairs of sequences of words, both in the same language, that have the same meaning in at least some contexts. Given a text, "paraphrasing" is the act of generating an alternate sequence of words that conveys the same meaning. Since the meaning of a text is determined only when its context is given, paraphrases are sometimes referred to as "dynamic translations" or "semantic equivalents". Identifying paraphrases is an important capability for many natural language processing applications, including machine translation, as a possible workaround for the problem of limited coverage inherent in a corpus-based translation approach (Callison-Burch et al., 2006; Marton et el., 2009). Other applications of paraphrasing include automatic evaluation of summaries (Zhao et al., 2008) and question answering (Duboue and Chu-Carroll, 2006; Riezler et al., 2007).

There are two main directions of work on paraphrases that one can find in this field: investigating an automatic approach for uncovering paraphrases in a given corpus and using paraphrases to improve the performance of a specific task. In this paper, we introduce a novel method for extracting Arabic paraphrases from a corpus of comparable documents as part of our work on improving Arabic-to-English machine translation. *Comparable* documents are ones that deal with the same topic, such as two newspaper reports of the same event. The extraction technique is based on a learning method, known as "co-training" (Blum and Mitchel, 1998) and inspired by the work of Barzilay and McKeown (2001) for finding paraphrases in a parallel monolingual English corpus. In order to validate our technique, we have applied it also to a similar English corpus.

Like many other Semitic languages, Arabic is highly inflected; therefore, data sparseness becomes even more noticeable than in English and extracting paraphrases from a corpus turns out to be even more complicated. Arabic words are derived from a root and a pattern (template), combined with prefixes, suffixes and circumfixes. Using the same root with different patterns may yield words with different meanings. Words are inflected for person, number and gender; prefixes and suffixes are then added to indicate definiteness, conjunction, various prepositions and possessive forms. We will list some of the morpho-syntactic features we use for identifying paraphrases in the corpus. Based on the definition, paraphrases are identified as part of the context they are mentioned in within the corpus. Paraphrase is in fact only one of the semantic relations that can be identified to hold between two word sequences in their contexts; it can be seen as a special case of textual entailment (Dagan and Glickman, 2004), where each sequence entails the other.

There are several existing approaches for inferring paraphrases from a corpus, which differ from one another in the type of corpus they employ. Some require bilingual parallel corpora (Callison-Burch et al., 2006; Zhao et el., 2008), some need monolingual parallel corpora (Barzilai and McKeown, 2001), some need general monolingual corpora (Marton et al., 2009) and others need corpora of comparable documents (Rui and Callison-Burch, 2011; Dolan et al., 2004). Bilingual parallel corpora, pairing Arabic with languages other than English, are very hard to obtain. In this paper we take the last approach, leaving the other directions for future investigation.

Section 2 cites some related work. Our proposal is described in Section 3 with some experimental results reported in Section 4. Conclusions are given in the last section.

2 Related work

To the best of our knowledge, ours is the first work on paraphrasing in Arabic. In our previous work (Bar and Dershowitz, 2010), we extracted Arabic synonyms, that is, single-word paraphrases, using the English glosses provided by SAMA (Maamouri, 2010), along with WordNet for English (Fellbaum, 1998). The inferred synonyms were used to improve a corpus-based translation system. Salloum and Habash (2011) developed a rule-based algorithm for generating Modern Standard Arabic (MSA) paraphrases for dialectical Arabic phrases given to a statistics-based automatic translation system. They focused only on input phrases that do not exist in the translation table used by the translation system, for the purpose of improving its coverage. The MSA paraphrases were generated mostly using different morphological variations of the input words. They reported a slight improvement in BLEU score (Papineni, 2002) over a baseline system that does not use their generated paraphrases. In another work, by Denkowski et al. (2010), 726 Arabic paraphrases were manually generated and confirmed using Amazon's Mechanical Turk, from the NIST OpenMT 2002 development set (Garofolo, 2002). That was mainly done with the purpose of improving the evaluation of an English-to-Arabic machine translation system.

There are also related works in other languages. We only mention a few. Marton et al. (2009) found paraphrases to improve Spanish-to-English and English-to-Chinese statistical machine translation (SMT). For each phrase (as defined in SMT) that was left without a translation, they looked for it in a monolingual corpus and recorded the contexts in which it appeared. They modeled the contexts using a vector that captured phrase occurrences with their context words, and searched for other phrases with the most similar vector of occurrences to improve the translation. Callison-Burch et al. (2006) measured the effect of using paraphrases on Spanish-to-English and French-to-English SMT. They reported a significant improvement in coverage and in the final translation. The paraphrases were automatically extracted following the technique developed by Bannard and Callison-Burch (2005), using several parallel corpora of French and Spanish paired with other languages. This method is usually referred to as "pivoting". Both of these works claimed improvements in the translations. Callison-Burch (2008) and Zhao et al. (2008) developed this approach further by adding syntactic constraints to the extraction algorithms. In a recent work by Wang and Callison-Burch (2011), English paraphrases were found in a corpus of comparable documents. Similar to what we have done, they started with a large English corpus to find comparable documents. Those documents were used to find comparable sentences from which they extracted sub-sentential comparable fragments, that is, paraphrases. They used a chunker for finding linguistically safe boundaries for the fragments they extracted, and matched fragments based on the *n*-gram alignment method.

The most inspiring work for us is the one by Barzilay and McKeown (2001), in which paraphrases are extracted from a corpus containing multiple English translations of the same source. Using this type of corpus allow them to mark initial aligned anchors, chosen based on the results of an alignment algorithm, and to train a classifier to identify the best context environments surrounding potential paraphrases. Based on the resulting contexts, another classifier was trained for finding new paraphrases. This "co-training" process was repeated until no new paraphrases were extracted. In our work, we follow the same idea, implemented on Arabic. Since there is no monolingual parallel corpus available for Arabic, we created a corpus of comparable documents and used it as a resource for paraphrasing. Considering that Arabic is a

morphologically rich language, we incorporated morphological features of the surrounding words as well as the paraphrase patterns themselves.

3 Inferring paraphrases

3.1 Preparing the corpus

As just mentioned, our approach to inferring paraphrases is based on the work of Barzilay and McKeown (2001) on finding paraphrases in different English translations of the same source text. Understanding how powerful such a resource can be for paraphrasing, but finding no such resource for Arabic, we built a corpus of comparable documents, that is, distinct documents dealing with the same topic or event. This corpus was extracted from the Arabic Gigaword 4.0 (Parker, 2009), which contains newswire documents published by several news agencies, grouped by their publication date. Pairing documents, based on their topic, was done automatically using cosine similarity over the lemma-frequency vector of every document, with the lemma of every word extracted using MADA (Habash and Rambow, 2005; Roth et al., 2008). We considered candidates for document pairs only when they were published by different news agencies on the same day. For every document published by one agency, we pair it with a document from the agency that maximizes the similarity score over all the other documents published by the same agency on the same day. Not only that, we require that the score be higher than a predefined threshold that was set, in our experiment settings, to make sure that every candidate pair is composed of two documents sharing at least one third of the largest one. We also tried using lower thresholds for which we retrieved additional pairs; however, precision decreased linearly. It is obvious, then, that this approach prefers precision to recall; in other words, we probably miss a large number of potential candidates, while the candidates that we do extract are likely correct.

All together, we created 690 document pairs, comprising about half a million words. Our corpus of comparable documents was manually evaluated by two Arabic speakers. We randomly selected 120 document pairs out of the 690 and, for each, asked the evaluators for a simple "yes" or "no" answer to the question, "Do both documents discuss the same event?" The results are encouraging: out of the 120 pairs, 100 were classified as correct by both evaluators. Of the other 20 instances, 5 were classified "yes" by one evaluator. The rest of the pairs actually dealt with the same general domain but were not specifically discussing the same event. This positive evaluation allowed us to use this corpus in the next step of our inference technique.

Every document was pre-processed with AMIRAN before being given to the inference classifier, described in the next section. AMIRAN, an updated version of the AMIRA tools (Diab et al., 2004, 2007), is a tool for finding the context-sensitive morpho-syntactic information. AMIRAN combines AMIRA output with morphological analyses provided by SAMA. AMIRAN is also enriched with Named-Entity-Recognition (NER) class tags provided by (Benajiba et al., 2008). For every word, AMIRAN is capable of identifying the clitics, diacritized lemma, stem, full part-of-speech tag (excluding case and mood), base-phrase chunks and NER tags. The corpus is obviously not annotated with paraphrasing-related information and there is no alignment indication included at any level.

3.2 Inference technique

To infer new paraphrases from the corpus, we follow the "co-training" technique, training two different classifiers: one for modeling the context of a potential paraphrase and another for modeling the features of the paraphrase pattern itself. The main idea of the co-training approach applied to unlabeled data is to use the two classifiers on different views of the data. In our case, the two views are the context (CX) and the pattern (PT), with one classifier labeling the most reliable unlabeled data items for training the second classifier. Then, the second classifier can label some of the data items for training the first one. This process is repeated several times, and the labeled data collected during the entire run is returned. The algorithm runs in iterations; each iteration increases the number of words a potential paraphrase may contain, that is, in the first iteration only single-word paraphrases are allowed to be found, in the second one, paraphrases composed of up to two words are allowed, and so on. The input of the algorithm is the pairs of documents that we found on the previous section, from which we extract pairs of word sequences. A *pair of word sequences* is composed of two sequences, one from each of a pair of comparable documents. Since alignment at any level does not exist for comparable documents, we consider all the possible pairs of word sequences, given a pair of documents. To avoid too much noise, we restrict a word sequence for consideration to be composed of at least one non-function word and it to not break a base-phrase in the middle, similar to (Wang and Callison-Burch, 2011). Function words, in our case, are identified based on their part-of-speech and base-phrase tags, as provided by AMIRAN. Otherwise, a huge number of pairs containing only function words, not too important for paraphrasing, would be considered. The number of iterations, and concomitantly, the maximum length of the output sequences, is a parameter we control. As implied before, we start with single-word sequences and increase this parameter with every iteration. During the entire run of the algorithm, we maintain two sets of pairs of word sequences:

1. Labeled – containing pairs of word sequences with their label, "true" to indicate paraphrases and "false" to indicate that the word sequences are not paraphrases of each other. This set starts off empty.
2. Unlabeled – containing pairs of word sequences that are still waiting for their label assignment by the algorithm.

In every iteration, the algorithm performs the following steps:

1. deterministic labeling of potential paraphrases;
2. training the CX classifier using the labeled set as training data;
3. running CX on unlabeled pairs and labeling the most reliable ones;
4. training the PT classifier using the labeled set as training data;
5. running the PT classifier on the labeled set;
6. labeling some unlabeled pairs, based on the labels provided by both classifiers.

We now describe these steps in greater detail.

We cannot estimate in advance the weight of the selected features and their effect on the predictions of the classifiers; therefore, we chose to use support vector machine (SVM) classifiers (Vapnik and Cortes, 1995) because of their good generalization property. Technically, the classifiers are trained on the WEKA platform (Hall et al., 2009) running with the LibSVM library (Chang and Lin, 2011). One drawback of using SVM in this kind of setting is the long running

time of the training algorithm. Because we are running the trainer twice during every iteration, this drawback becomes even more pronounced.

The labeled pairs are used as training data for both classifiers, with every pair formatted as a feature vector. The features for the CX classifier capture some morpho-syntactic information expressed by the window-based context words. In the current experiment, we use a window of size three, that is, three words before each word sequence from the pair, and three words afterward. That gives us twelve words from which we extract features for representing a single pair and that number does not change during the entire learning process. Table 1 shows an example for a context.[1] The two main columns represent two Arabic sentences with their corresponding English translations, for easy reading. The emphasized texts are the actual paraphrases while the surrounding words are composing the context, which is described in the last row. In this case, the paraphrase pair is composed of a single identical lemma, inflected differently for person. The context of a paraphrase pair is composed of four parts: left and right words of each of the paired texts.

	Sentence 1	**Sentence 2**
Sentence	مكتب السنيورة وديوان ألمرط ينفيان خبرا عن لقاء في شرم الشيخ	مكتب السنيورة ينفي خبرا عن لقائه مسؤولين إسرائيليين
Transliteration	*mktb Alsnywrp wdywAn >wlmrt* ***ynfyAn*** *xbrA En lqA' fy $rm Al$yx*	*mktb Alsnywrp* ***ynfy*** *xbrA En lqA}h ms&wlyn <srA}ylyyn*
Translation	Seniora's office and the Olmert administration **deny** a story about a meeting in Sharm al-Sheikh	Seniora's office **denies** a story about his meeting with Israeli officials
Context	*Alsnywrp wdywAn >wlmrt* […] *xbrA En lqA'*	*mktb Alsnywrp* […] *xbrA En lqA}h*

TABLE 1 – An example for a context. The word sequence (here of size one) is highlighted in boldface.

The PT classifier makes its predictions based on the word sequences themselves; their number varies as the iteration number increases. For both classifiers, we use a quadratic kernel for capturing the common effect of all the features on prediction. Tables 2 and 3 summarize the features we currently use for building the feature vectors for the CX and PT classifiers, respectively. NER tags are assigned to persons, organizations, geo-political organizations and locations. The gloss-match rate is calculated for both sides of the context. In the example of Table 1, there is no word that matches on the left side (note that proper nouns usually do not have glosses). However, on the right side خبرا عن لقاء *xbrA En lqA'* ("a story about a meeting") matches خبرا عن لقائه *xbrA En lqA}h* ("a story about his meeting") with all three words on the gloss level; therefore, the left gloss-match rate is 0 and the right one is 1. The same calculation works with lemma-match rates on the lemma level. The morphological features we currently use in the PT classifier capture some common Arabic morphological variations. They are all Boolean values indicating whether the word expresses the feature or not. For example, the word وبكتابه *wbktAbh* ("and in his book") expresses conjunction, preposition and possessive. When working with Arabic, a highly inflected language, morphological features may contribute to the classification

[1] We use the Buckwalter transliteration scheme (Buckwalter, 2002) for rendering Arabic script in Romanization throughout this paper.

performance. We intend to further explore this direction in the future. The *n*-gram score is a simple language model score for capturing the co-occurrence of the candidate sequence words.

Feature	Description
Lemma, POS, NER, BP	of each context word
Gloss-match rate	The rate of gloss match on each side of the context (left and right)
Lemma-match rate	The rate of lemma match on each side of the context

TABLE 2 – The features we use for training the CX classifier on Arabic.

Feature	Description
n-gram score	Normalized *n*-gram frequency score for word sequences up to 4 words (2-4 grams)
POS, NER, BP	of each sequence word
Boolean morphological features (exists / does not exist): Conjunction, Possessive, Determiner and Prepositions	of each sequence word
Sequence length	The number of words in each sequence

TABLE 3 – The features we use for training the PT classifier on Arabic.

The first time one of the classifiers is trained, it needs some labeled items. With "co-training", those items are usually provided by manual annotation of a relatively small fraction of the data or, in this case, by using an automatic deterministic annotation algorithm. Therefore, in the first step of every iteration, the algorithm enriches the labeled set with additional "true" labeled pairs following a deterministic approach. Since it is very difficult to obtain a word or sentence-level alignment of two given comparable documents, our algorithm simply adds all the pairs whose word sequences match on the lemma level, word by word. If the lemma does not exist, we use the word's surface form for matching. Lengths of word sequences are determined by the iteration number, so in the first iteration only sequences of size 1 are added, in the second iteration sequences of size 2 are added, and so forth. Such a pair, matched on the lemma level, is shown in Table 1. Note that paraphrases work on the sense level, rather than on the surface form; however, our assumption is that, because we are using sequences from comparable documents, their senses may be the same with a reasonable high probability. Note that, since we are using the context-sensitive lemmas for matching, one can think of that as matching words on the sense level. However, AMIRAN was trained mostly with morpho-syntactic features and therefore achieves good performance in identifying the common lemma of a context-sensitive part-of-speech tag for every word. When a word may have two or more different lemmas for the same part-of-speech tag that have different senses, AMIRAN does not perform as well. For example, the word أمانة

>mAnp has three different noun lemmas: *>amAnap_1* (“faithfulness”), *>amAnap_2* (“secretariat”) and *>amAnap_3* (“deposit”).

That approach leaves us with some deterministically selected “true” examples; however, it does not provide us with the necessary “false” examples. In the first iteration, we consider word sequences of size 1 only. Our assumption is that word pairs sharing some of their senses in common may be considered paraphrases, thus cannot be naturally selected as “false” examples. Currently we use the English gloss of every word, as provided by AMIRAN, to select word pairs with different gloss values as “false” examples. Therefore, under this condition, the Arabic word pair مجال *mjAl* and منطقة *mnTqp* is not considered as a “false” example because they share the same gloss value: “area”. An alternative approach, which we plan to employ in the future, would be using Arabic WordNet (Black et al., 2006). It implies that, in our first iteration, only word pairs that have the same English gloss and not the same Arabic lemma are put in the unlabeled set. That dramatically reduces the amount of paraphrases of size one, better known as “synonyms”, that we can find. Since we are more interested in longer paraphrases, we can live with this limitation.

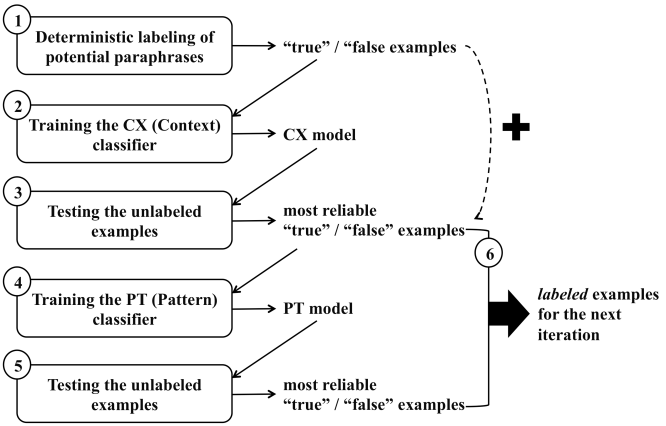

FIGURE 1 – An overview of the paraphrase inference co-training algorithm.

In subsequent iterations, “false” examples will be assigned automatically by the classifiers of the previous one, in the following way: after training the CX classifier in step 2, we use the classifier to tag the unlabeled pairs in step 3. Some pairs are assigned with the “true” label and some with “false”. Those for which the classifier has a “good sense” are added to the labeled set with their corresponding label. “Good sense” is measured with a confidence score that is provided by LibSVM along with every tested pair. Since this score is based on margin length calculations, one should use them carefully. Currently, we only set some threshold values for adding pairs to the labeled set, with a high score, empirically determined. The unlabeled set is also updated with additional examples of length not exceeding the iteration number. In that sense, the iteration number is actually an upper bound on the length of the examples, allowing the algorithm to select sequences of a lower length paired with longer sequences. For example, in the second iteration, the unlabeled set also contains examples that pair a sequence of one word with a sequence of two

words. The labeled set after step 3 contains "true" as well as "false" pairs, added by both the deterministic algorithm and the CT classifier, for training the PT classifier. Steps 4 and 5 train and test the classifier PT on the labeled and unlabeled sets respectively. Finally, in step 6, pairs that receive the same label from both classifiers, with a confidence score higher than the predefined threshold, are added to the labeled set with their corresponding label and stay there forever. This labeled set is used as part of the training data in the next iteration. The number of iterations is manually configured upon initialization of the algorithm and at the end, the "true" pairs are deemed paraphrases. The entire process is summarized in Figure 1.

To get a feeling for the robustness of the methodology, we applied the same technique to the task of generating paraphrases in English. English has shallow morphology as compared with Arabic, on one hand, but, on the other hand, uses more words than Arabic to convey the same meaning. Based on this observation, for English, we changed the settings of the data for using a window of size 4 instead of 3 and removed most of the morphology-related features. The English set of features for the CX and PT classifiers are summarized in Tables 4 and 5, respectively.

Feature	Description
Lemma, POS, NER, BP	of each context word
Lemma-match rate	The rate of lemma match on each side of the context

TABLE 4 – The features we use for training the CX classifier on English.

Feature	Description
n-gram score	Normalized *n*-gram frequency score for word sequences up to 4 words (2-4 grams)
POS, NER, BP	of each sequence word
Possessive form	of each sequence word
Sequence length	The number of words in each sequence

TABLE 5 – The features we use for training the PT classifier on English.

Comparable documents were extracted using the same technique from a relatively small part of the English Gigaword (5th ed.) (Parker et al., 2011). We preprocessed the documents using the OpenNLP library. For every word, we determined its part-of-speech, base-phrase and named-entity tags. The lemma of each word was retrieved from WordNet (Fellbaum et al., 1998) by providing it with the surface form and the part-of-speech tag as inferred by OpenNLP. Overall, we found 294 document pairs, containing about 220,000 words. Similar to the evaluation step of the Arabic corpus, we randomly selected 80 document pairs for vetting their correspondence to each other. Out of the selected 80 document pairs, 65 were classified as "yes" instances by both evaluators. Of the other 15 instances, 3 were classified as "yes" by only one evaluator. As for Arabic, the rest of the pairs were actually dealing with the same general domain but not specifically discussing the same event. The inference algorithm for English worked exactly as described above. Recall that in the first iteration on Arabic, we used a deterministic algorithm for labeling some of the data for training the classifiers for the first time. For Arabic, we used the

English gloss values of the Arabic words for finding "false" examples; for English, we use WordNet for the same task in such a way that synonyms are not considered as "false" examples.

4 Results and evaluation

4.1 Experiment settings

Our initial experiments perform only five iterations on both corpora (Arabic, as well as English), which means that we find paraphrases of no longer than five words. The two classifiers are configured with different thresholds. The confidence score given by LibSVM for every classification is a value between 0 and 1; therefore, we experimented with different threshold values and realized that the best settings in this case are obtained when using 0.85 for "true" pairs for the CX classifier and 0.75 for the PT classifier. For the "false" pairs, we use 0.75 for both classifiers. Since we noticed that the number of "false" pairs is much larger than the number of "true" ones in the training data of every iteration, we defined another parameter (currently 6) that limits the factor of "false" pairs allowed in the training data with respect to the "true" pairs.

In the next section, we show some results when running over 240 document pairs in Arabic, containing about 165,000 words, and 40 English document pairs containing about 11,000 words.

4.2 Results

First, we give some statistics on the results obtained by the inference algorithm on both the Arabic and English corpora, in Tables 6 and 7, respectively.

	"false" pairs	**"true" pairs**	**Unique paraphrase pairs**	**Unlabeled pairs**
Initialization	22,885,104	66,317		19,480
After iteration 1	23,799,787	(+1,726) 68,043		3,166,935
After iteration 2	24,759,791	(+3,757) 71,800	954	2,790,574
After iteration 3	25,349,489	(+2,623) 74,423	416	2,198,253
After iteration 4	26,221,889	(+451) 74,874	331	1,557,931
After iteration 5	26,900,833	(+101) 74,975	72	878,987
Total			**1,773**	

TABLE 6 – Statistics and final results of the inference algorithm running on the Arabic corpus.

In both tables, the initialization row shows the number of "true" and "false" examples as was labeled by the deterministic algorithm and the size of the unlabeled examples set. In the following rows, the numbers refer to the results of the specific iteration. The numbers of "true" and "false" pairs reported on every line are the aggregated numbers collected from all previous iterations. Recall that at the beginning of every iteration, a deterministic algorithm adds pairs of word sequences that match on the lemma level, word by word; hence, the number of "true" pairs in every line is the sum of the pairs from the previous iterations, the pairs added by the deterministic algorithm for the next iteration and the paraphrase pairs inferred by the current iteration. The third column, unique paraphrase pairs, is merely the number of unique paraphrase pairs inferred during the current iteration. The parenthesized numbers indicate the difference in

the quantity of "true" pairs from the previous iteration. So, the total number of extracted paraphrases is the number written on the total line in the unique paraphrases column. In Arabic, we found 1,773 paraphrase pairs and in English we found 525. This process can be scaled up for finding more paraphrases. We do not include paraphrases generated after the first iteration because, by definition, they are composed of synonymous words. Recall that, during initialization, the deterministic algorithm adds pairs to the unlabeled set if their paired words are synonyms in English or share the same English gloss, in Arabic. Table 8 shows some statistics for the entire inference process.

	"false" pairs	**"true" pairs**	**Unique paraphrase pairs**	**Unlabeled pairs**
Initialization	876,947	32,972		3,597
After iteration 1	960,840	(+868) 33,840		86,648
After iteration 2	1,058,970	(+1,633) 35,473	230	58,312
After iteration 3	1,109, 746	(+1,194) 36,667	177	21,332
After iteration 4	1,127,643	(+339) 37,006	94	6,677
After iteration 5	1,128,475	(+52) 37,058	24	1,490
Total			**525**	

TABLE 7 – Statistics and final results of the inference algorithm running on the English corpus.

The raw data corpus size is a rough estimation of the amount of words we had in the corpus at the beginning. Note that currently we did not use the entire Gigaword corpora: in Arabic we used about 30% of the entire set and in English we only used about 10% of the documents. The following column shows the number of comparable document pairs we found using the pairing algorithm described above. Since the pairing algorithm was designed to prefer recall over precision, the number of comparable documents is lower than might be expected considering the relatively large number of words we had in the raw corpus. We expect that this number will grow larger once we improve the pairing algorithm. The next column, number of words used in inference, sums up the number of words of the entire set of comparable document pairs from the previous column. The last column shows the number of paraphrase pairs extracted by the inference algorithm.

	Raw data corpus size	**Extracted comparable document pairs**	**Comparable documents used in inference**	**Number of words used in inference**	**Number of inferred unique paraphrases**
Arabic	~20,000,000	690	240	165,369	1,773
English	~1,000,000	294	40	11,600	525

TABLE 8 – General statistics on the entire inference process.

Comparing the results to the results retrieved by other works is difficult because there is neither a shared task for paraphrase extraction nor common resources for comparison. Therefore, we show some manual evaluations of our results. The evaluation was performed by two Arabic-English speakers by going over the reported paraphrases one by one. For each pair, we assigned one label: *P* – indicating correct paraphrase, *E* – indicating unidirectional entailment, *R* – related (for

other semantic relations except antonyms, e.g. San Diego/Los Angeles) and *F* – wrong (including antonyms). Table 9 and 10 summarizes our preliminary evaluation report on Arabic and English, respectively.

Length	Evaluated	*P*	*E*	*R*	*F*	Precision
2	120	49	12	25	34	71%
3	95	45	10	11	31	69%
4	70	26	4	5	35	50%
5	50	24	2	7	20	66%
Total	335	144	28	48	120	**66%**

TABLE 9 – Manual evaluation summary for Arabic. *P*: paraphrases, *E*: unidirectional entailment, *R*: related, *F*: wrong, i.e. unrelated or antonyms.

The evaluation results reported in both tables are based on the agreement of the two evaluators; in other words, we report here only on pairs that were annotated by both evaluators with the same tag. Note that the first column, length, indicates the number of words of the largest phrase included in the evaluated paraphrase pair. Paraphrase pairs containing a single word in both phrases were not evaluated at all. In the last column, we calculate the precision, considering pairs tagged with *P*, *E* and *R* as positive instances. The last row summarizes the results. In Arabic, 66% of the generated paraphrase pairs are at least considered as semantically related; among them, about 43% are considered real paraphrases. In English, only 63% of the paraphrase pairs are considered related, out of which 30% are real paraphrases. As can be seen from the tables, there is no preferred length for the inference algorithm. We see a slight improvement in the precision of paraphrases up to length three; however, this improvement does not seem significant, considering the relatively small amount of evaluated pairs.

When we increased the threshold on the confidence that is used by the PT classifier on English to 0.9, the number of paraphrases reported by the inference algorithm decreased to 330 and the average number of similar words in a pair, increased. As a results of that, the overall precision got improved to 72%, calculated over 250 evaluated pairs. These results helped us understand the effect of the PT classifier on performance. The pairs with a high confidence score, as reported by the PT classifier, are most likely to be real paraphrases; however, in most cases, the word sequences of such a pair share more words in common than do other pairs (e.g. “the U.S. Air Forces” ⇔ “the United States Air Force”).

Length	Evaluated	*P*	*E*	*R*	*F*	Precision
2	120	23	11	37	49	59%
3	60	28	6	9	17	71%
4	50	15	8	8	21	62%
5	25	8	5	2	10	60%
Total	255	74	30	56	97	**63%**

TABLE 10 – Manual evaluation results for English. *P*: paraphrases, *E*: unidirectional entailment, *R*: related, *F*: wrong, i.e. unrelated or antonyms.

Table 11 gives some examples of Arabic, as well as English, paraphrase pairs that were extracted by our inference algorithm. The third column is the evaluation score given by one of the evaluators.

Language	Paraphrase pair	Evaluation score
Arabic	الرئيس الفلسطيني *Alr}ys AlflsTyny* (“the Palestinian president”) ⇔ السلطة الوطنية الفلسطينية *AlslTp AlwTnyp AlflsTynyp* (“the Palestinian authority”)	Related
Arabic	جورج ووكر بوش *jwrj wwkr bw$* (“George Walker Bush”) ⇔ جورج بوش *jwrj bw$* (“George Bush”)	Paraphrases
Arabic	المؤتمر السادس *Alm&tmr AlsAds* (“the Sixth conference”) ⇔ الاجتماع الوزاري السادس *AlAjtmAE AlwzAry AlsAds* (“the Sixth ministerial meeting”)	Paraphrases
Arabic	دانييل جلازر *dAnyyl jlAzr* (“Daniel Glaser”) ⇔ دانيئل غلاسر *dAny}l glAsr* (“Daniel Glaser”)	Paraphrases
Arabic	كيلي غونزاليز وانجولو *kyly gwnzAlyz wAngwlw* (“Kaylie Gonzales and Angelo”) ⇨ الارجنتينيين الخطيرين *AlArjntynyyn AlxTyryn* (“the dangerous Argentinians”)	Unidirectional entailment
Arabic	البرلمان الجديد *AlbrlmAn Aljdyd* (“the new Parliament”) ⇔ المجلس الوطني السابع عشر *Almjls AlwTny AlsAbE E$r* (“the Seventeenth Parliament”)	Paraphrases
Arabic	الحدود السورية اللبنانية *AlHdwd Alswryp AllbnAnyp* (“the Syrian-Lebanese borders”) ⇔ الحدود السورية *AlHdwd Alswryp* (“the Syrian border”)	Unidirectional entailment
English	could veto ⇔ threatened to veto	Related
English	the U.S. Naval Task Force ⇔ a US Naval Task Group	Paraphrases

English	Beijing's policy ⇔ the China's policy	Paraphrases
English	a poor and little-developed province ⇔ its resource-rich northwestern province	Wrong
English	U.S. beef and related products ⇨ beef products	Unidirectional entailment
English	a magnitude 6.0 earthquake ⇨ the quiver	Unidirectional entailment
English	will only endanger ⇔ will not only endanger	Wrong

TABLE 11 – Example results in both Arabic and English.

Conclusions

The method suggested here has demonstrated its potential for inferring paraphrases from a corpus of comparable documents, using "co-training". As we have seen, incorporating morphological features for a highly inflected language, such as Arabic, is very effective. SVM with its generalization property was a natural option for dealing with combination of features that can play an important role for identifying paraphrases. Finding more features that help to properly match the true senses of word sequences is definitely a direction for future investigation. In a similar experiment performed on English, we still saw encouraging results, despite the smaller corpus. In the next stage of research, we plan to scale up the experiments and use more raw data along with an improved document-pairing algorithm for inferring additional paraphrases. We also plan to use those paraphrases within an Arabic-to-English translation system so as to hopefully improve the quality of the translations.

References

Bannard, C. and Callison-Burch, C. (2005). Paraphrasing with bilingual parallel corpora. In *Proceedings of the Annual Meeting of Association of Computational Linguistics (ACL 2005)*, pages 597-604, Ann Arbor, MI.

Bar, K. and Dershowitz, N. (2010). Using synonyms for Arabic-to-English example-based translation. In *Proceedings of the Ninth Conference of the Association for Machine Translation in the Americas (AMTA 9)*, Denver, CO.

Barzilay, R. and McKeown, K. (2001). Extracting paraphrases from a parallel corpus. In *Proceedings of the 39th Annual Meeting of Association of Computational Linguistics (ACL 2001)*, pages 50-57, Toulouse, France.

Benajiba, Y., Diab, M., and Rosso, P. (2008). Arabic named entity recognition: An SVM-based approach. In *Proceedings of the Arab International Conference on Information Technology (ACIT-2008)*, Hammamet, Tunisia.

Black, W., Elkateb, S., Rodriguez, H., Alkhalifa, M., Vossen, P., Pease, A., and Fellbaum, C. (2006). Introducing the Arabic WordNet project. In *Proceedings of the 3rd Global Wordnet Conference (GWC 2006)*, pages 295-299, Jeju Island, South Korea.

Blum, A. and Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. In *Proceedings of 11th Annual Conference on Computational Learning Theory (COLT)*, pages 92–100, Madison, WI.

Buckwalter, T. (2002). Buckwalter Arabic morphological analyzer Version 1.0. LDC catalog number LDC2002L49.

Callison-Burch, C. (2008). Syntactic constraints on paraphrases extracted from parallel corpora. In *Proceedings of the Conference for Empirical Methods in Natural Language Processing (EMNLP 2008)*, pages 196-205, Honolulu, HI.

Callison-Burch, C., Koehn, P., and Osborne, M. (2006). Improved statistical machine translation using paraphrases. In *Proceedings of the North American Association for Computational Linguistics (NAACL 2006)*, New York City, NY.

Chang, C. C. and Lin, C. J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1-27:27. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

Dagan, I. and Glickman, O. (2004). Probabilistic textual entailment: Generic applied modeling of language variability. In *PASCAL workshop on Learning Methods for Text Understanding and Mining*, pages 26-29, Grenoble, France.

Denkowski, M., Al-Haj, H., and Lavie, A. (2010). Turker-assisted paraphrasing for English-Arabic machine translation. In *Proceedings of the Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk at the Conference of Human Language Technology-North American Association for Computational Linguistics (HLT-NAACL 2010),* pages 66-70, Los Angeles, CA.

Diab, M. (2009). Second generation tools (AMIRA 2.0): Fast and robust tokenization, POS tagging, and base phrase chunking. *MEDAR 2nd International Conference on Arabic Language Resources and Tools*, Cairo, Egypt.

Diab, M., Hacioglu, K., and Jurafsky, D. (2004). Automatic tagging of Arabic text: From raw text to base phrase chunks. In *Proceedings of Human Language Technology-North American Association for Computational Linguistics (HLT-NAACL 2004)*, pages 149-152, Boston, MA.

Dolan, B., Quirk, C., and Brockett, C. (2004). Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING 2004),* Geneva, Switzerland.

Duboue, P. A. and Chu-Carroll, J. (2006). Answering the question you wish they had asked: The impact of paraphrasing for question answering. In *Proceedings of Human Language Technology-North American Association for Computational Linguistics (HLT-NAACL 2006)*, pages 33-36, New York City, NY.

Fellbaum, C. (1998). *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, MA.

Garofolo, J. (2002). NIST OpenMT Eval. http://www.itl.nist.gov/iad/mig/tests/mt/2002/.

Habash, N. and Rambow, O. (2005). Arabic tokenization, morphological analysis, and part-of-speech tagging in one fell swoop. In *Proceedings of the Conference of American Association for Computational Linguistics*, pages 578-580, Ann Arbor, MI.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. (2009). The WEKA data mining software: An update. *SIGKDD Explorations*, volume 11, issue 1.

Maamouri, M., Graff, D., Bouziri, B., Krouna, S., Bies, A., and Kulick, S. (2010). Standard Arabic morphological analyzer (SAMA), Version 3.1. *Linguistic Data Consortium*, Philadelphia, PA.

Marton, Y., Callison-Burch, C., and Resnik, P. (2009). Improved statistical machine translation using monolingually-derived paraphrases. In *Proceedings of the Conference for Empirical Methods in Natural Language Processing (EMNLP 2009)*, Singapore.

Papineni, K., Roukos, S., Ward, T., and Zhu, W. J. (2002). Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the ACL 40th Annual Meeting*, pages 311-318, Philadelphia, PA.

Parker, R., Graff, D., Chen, K., Kong, J., and Maeda, K. (2009). Arabic Gigaword, Fourth Edition. *Linguistic Data Consortium*, Philadelphia, PA.

Parker, R., Graff, D., Chen, K., Kong, J., and Maeda, K. (2011). English Gigaword, Fifth Edition, *Linguistic Data Consortium*, Philadelphia, PA.

Riezler, S., Vasserman, A., Tsochantaridis, I., Mittal, V., and Liu, Y. (2007). Statistical machine translation for query expansion in answer retrieval. In *Proceedings of Association for Computational Linguistics (ACL 2007)*, pages 464-471, Prague, Czech Repulic.

Roth, R., Rambow, O., Habash, N., Diab, M., and Rudin, C. (2008). Arabic morphological tagging, diacritization, and lemmatization using lexeme models and feature ranking. In *Proceedings of Association for Computational Linguistics (ACL 2008)*, pages 117-120, Columbus, Ohio.

Salloum, W. and Habash, N. (2011). Dialectal to standard Arabic paraphrasing to improve Arabic-English statistical machine translation. In *Proceedings of the Dialects workshop at the Conference for Empirical Methods in Natural Language Processing (EMNLP 2011)*. Edinburgh, UK.

Vapnik, V. and Cortes, C. (1995). Support vector networks. *Machine Learning*, vol. 20, pages 273-297.

Wang, R. and Callison-Burch, C. (2011). Paraphrase fragment extraction from monolingual comparable corpora. In *Proceedings of Fourth Workshop on Building and Using Comparable Corpora (BUCC)*, Istanbul, Turkey.

Zhao, S., Wang, H., Liu, T., and Li, S. (2008). Pivot approach for extracting paraphrase patterns from bilingual corpora. In *Proceedings of the Annual Meeting of Association of Computational Linguistics (ACL 2008*), pages 780-788, Columbus, OH.