Inferring Paraphrases for a Highly Inflected Language from a Monolingual Corpus

Kfir Bar, Nachum Dershowitz

School of Computer Science, Tel Aviv University, Ramat Aviv, Israel
{kfirbar,nachumd}@ post.tau.ac.il

Abstract. We suggest a new technique for deriving paraphrases from a monolingual corpus, supported by a relatively small set of comparable documents. Two somewhat similar phrases that each occur in one of a pair of documents dealing with the same incident are taken as potential paraphrases, which are evaluated based on the contexts in which they appear in the larger monolingual corpus. We apply this technique to Arabic, a highly inflected language, for improving an Arabic-to-English statistical translation system. The paraphrases are provided to the translation system formatted as a word lattice, each assigned with a score reflecting its equivalence level. We experiment with the system on different configurations, resulting in encouraging results: our best system shows an increase of 1.73 (5.49%) in BLEU.

Keywords. Paraphrases, Arabic, Machine Translation

1 Introduction

Paraphrases are pairs of text fragments, both in the same language, that have the same meaning in at least one context. Given a text, "paraphrasing" is the act of generating an alternate phrase 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". Paraphrases are also recognized as a bidirectional textual-entailment relation [13]. 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 [11,29]. Other applications of paraphrasing include question answering [16,35,19] and automatic evaluation of summaries [40].

We usually distinguish between two levels of paraphrases: (1) *phrase* (sub-sentential) level refers to two variable-length text segments, each containing one or more words; and (2) *sentence* level, composed of two complete sentences.

We introduce a data-driven phrase-level paraphrasing technique and apply it to Arabic, a highly inflected language. The paraphrases are then employed to improve a phrase-based statistical translation system. The ideal setup for paraphrasing would probably be to have both a monolingual corpus and a bilingual parallel corpus as resources. However, since parallel corpora are not always available (for Arabic, there

are only ones paired with English), we use monolingual documents as the primary resource for our paraphrasing algorithm. In addition, bilingual unaligned comparable documents (not translations) are used to suggest paraphrases. The paraphrase pairs are generated automatically by extracting similar phrases, similar on the lemma level, each of which occurs in one of a comparable pair of documents. In this work, a pair of comparable documents is composed of two news-related articles that appear to cover the same story. Like other, similar, works that utilize monolingual corpora for paraphrasing (e.g. [29]), we focus on the context in which the phrases occur in the text, where the context of a phrase is represented by some of its preceding and following words. We train a classifier to identify paraphrases through their context, supervised by an initial set of automatically annotated pairs.

This work makes the following contributions:

1. It proposes a new paraphrasing technique for monolingual corpora, especially as applied to a highly inflected language.
2. It compares the translation quality obtained by using different types of paraphrases.
3. It shows how to improve translation quality by tuning with translated paraphrase lattices.

Like most other Semitic languages, Arabic is highly inflected; therefore, data sparseness is much more noticeable than in English, and extracting paraphrases from a corpus turns out to be even more complicated. Arabic words are inflected for person, number and gender; prefixes and suffixes are added to indicate definiteness, conjunction, prepositions and possessive forms. Due to Arabic's rich morphology, we work on the lemma level. Consequently, our "paraphrases" sometimes include pairs with shared meaning, ignoring their inflection for number, gender, and person. The motivation is that such pairs often have similar English renderings.

We proceed as follows: Section 2 reviews some relevant previous work. Section 3 describes the details of our new monolingual paraphrasing technique. It is followed by a section explaining how paraphrases help in translation. In Section 5, we provide some experimental results, and finally we conclude in Section 6.

2 Related Work

There are several data-driven approaches for paraphrasing, which may be divided by the type of corpora they use. Some use monolingual corpora (e.g., [27,29]), some use parallel corpora, either monolingual (e.g., [4,31]) or bilingual (e.g., [1,40]), and others use comparable documents (e.g., [5,34,15,39,2]).

Barzilay and McKeown [4] extracted English paraphrases from monolingual parallel corpora. They marked a few identical aligned words as anchors and treated them as potential paraphrases. Following the co-training approach [8], they trained two classifiers, one to model the environment surrounding potential paraphrases and another to model the characteristics of paraphrases' words. In a previous work [2] we adapted that technique to derive Arabic paraphrases from comparable documents. Although

we reported encouraging results, the quantity of paraphrases that particular technique can produce is severely limited, and as a result we believe that it would be most difficult to employ this technique on a large enough scale to make a significant difference for machine translation.

Bannard and Callison-Burch [1] used several bilingual parallel corpora of French and Spanish paired with other languages, a technique known as "pivoting", to find pairs of phrases in one language that translate similarly in one of the pivot languages. Using this technique, Callison-Burch et al. [11] showed an improvement in the translation quality generated by a phrase-based statistical translation system. Callison-Burch [10] and Zhao et al. [40] developed this approach further by adding syntactic constraints to the extraction algorithms. To the best of our knowledge, there are no available bilingual resources that pair Arabic with languages other than English and that are aligned on the sentence level.

Marton et al. [29] used a relatively large monolingual corpus for deriving paraphrases in unsupervised settings to improve a phrase-based statistical translation system. Generally speaking, potential paraphrases were found based on cosine similarity of their distributional profile that captures the occurrences of the phrases' surrounding words, modeled by log-likelihood [17]. They reported an improvement in translation quality when the system is using relatively small bilingual corpora. The quality of the resulted paraphrases is connected with the size of the monolingual corpus used by the algorithm. A relatively large amount of monolingual data is needed for calculating the statistics for the contextual words. Both Callison-Burch et al. [11] and Marton et al. [29] derived paraphrases only for unseen input phrases, that is, phrases that do not exist in the system's translation table. Jinhua et al. [24] considered paraphrases, derived by pivoting, also for input phrases that exist in the phrase table. They formatted the input text, augmented with paraphrases, as a word lattice [18] and showed that it outperformed a system that merely calculates paraphrases for unseen phrases. Inspired by that, we restructure the input sentence as a lattice and augment it with paraphrases of all the composing phrases, regardless of their presence in the phrase table. Furthermore, we show that tuning the system on such lattices helps improve the results. Nakov and Ng [30] employed a similar lattice technique to help improve the results of a Malay-to-English translation system by using Malay paraphrases of various sorts. (Malay is another morphologically rich language, mainly based on a derivational morphology, as opposed to the inflectional one in Arabic.)

Our experiments reconfirm the conclusion that paraphrasing aids translation (e.g., [11,29,24,30]), this time for Arabic.

3 Paraphrasing Technique

At its core, our paraphrasing technique takes inspiration from a number of the above prior works. In a previous work [2] we extracted some morpho-syntactic features from phrases' contextual words. We constructed pairs of phrases, each pair represented by a single vector containing the weights of their features, as extracted from both phrases. The pairs of phrases were extracted from comparable documents, simp-

ly by pairing every phrase from one document with all phrases from its comparable partner. We used a deterministic procedure to assign labels to some of the pairs indicating whether a pair is composed of paraphrases or not and employed co-training, using those pairs as an initial training set, to label the unlabeled pairs. The labeling procedure considered pairs of similar phrases as paraphrases, and pairs of single words that were not identified as synonyms by a simple thesaurus as negative examples. The co-training learning algorithm focused on the context of both phrases and the words within the two phrases. The drawback is that newly discovered paraphrases were extracted merely from comparable documents; therefore, their number was relatively low and highly dependent on the quantity of comparable documents used. Obtaining a large amount of comparable data, as needed in that work, should considered much more challenging than obtaining plain monolingual texts, as in this work.

Since we are interested in improving machine translation, our system takes a given phrase and looks for its paraphrases. The input is paired with candidate phrases extracted from a relatively large monolingual corpus. Phrases that share a similar context with that of the input phrase are deemed paraphrases. To measure similarity of contexts, we first train a binary classifier using a relatively small set of annotated pairs, extracted from comparable documents using a technique like [2]. The limited-size corpus of bilingual comparable documents is, however, only used for training purposes. Going beyond our use of morpho-syntactic features (such as part of speech tags and base-phrase chunks), and inspired by the distributional similarity approach taken by [29], we define a new feature for capturing the semantic similarity of contexts, by representing words based on their frequency and co-occurrence with the phrase they are surrounding. In contrast to [29], who use co-occurrences of words and phrases as the model for finding paraphrases, we use it as part of a larger set of features considered by the context classifier.

3.1 Training a Context Classifier

In building a context classifier, we attempt to learn similarities between the contexts of two phrases that are deemed paraphrases. A context in our case is modeled by features extracted from the surrounding words of each of the two phrases. The number of words may vary for each individual feature. In particular, the classifier is trained to handle a binary classification problem: given pairs of phrases, decide which pairs are paraphrases and which are not. To supervise the training process, we deterministically generate a learning set of positive and negative examples, provided with their contexts. Those are collected from pairs of comparable documents, that is, different news articles that cover the same story. Obtaining comparable documents is currently done by a simple automated technique [39,2] from Arabic Gigaword (4th ed.) [33], a corpus of newswire stories published by several news agencies and grouped by publication date. The documents were pre-processed by AMIRA 2.0 [14], so that every word is assigned its lemma, full part-of-speech tag (excluding case and mood), base phrase chunks and named-entity recognition (NER) tags [6]. Pairing documents based on topic was done using the lemma-frequency vector of every document, taking those with cosine similarity above a threshold set heuristically to prefer

precision to recall and considering only those articles that were published on the same day by different news agencies.

Given a pair of comparable documents, we begin by extracting all phrases (i.e., word sequences) of up to N words (here, N=6) from each document. We pair each phrase from one document with all the phrases from the other document, resulting in a relatively large set of pairs. Among those, we keep only those that we can label as positive or negative. A positive pair must comply with the following rules:

- Both phrases do not break a base phrase in the middle;
- both phrases contain at least one content word (non-functional, determined using the part-of-speech tag); and,
- both phrases match on the lemma level, word by word.

Since we work with words rather than senses, similar phrases do not always have the same meaning, given their local context. However, the fact that the phrases are taken from comparable documents suggests that they do share similar senses. Some positive examples are provided in Table 1.[1]

Table 1. Examples of positive phrase pairs.

different number	*wqAl AlmSdr* ↔ *wqAlt AlmSAdr* "and the source said" ↔ "and the sources said"
different proclitic	*bt$kyl Hkwmp* ↔ *wt$kyl Hkwmp* "with establishment of a government" ↔ "and establishment of a government"
exact match	*wzyr AlxArjyp* ↔ *wzyr AlxArjyp* "the minister for foreign affairs"

For negative examples, we select pairs of phrases that do not comply with the last rule and also not with one of the two others. This gives us enough confidence to believe that such phrase pairs are not paraphrases.

We use SVM [38] as the machinery for training the context classifier and employ a quadratic kernel, to enable the learning process to consider combinations of features. Technically, we used WEKA [23] as a framework combined with LibSVM [12]. The features that we use for training are described next.

3.2 Feature Extraction

We extract features from the contextual words surrounding each phrase. Given a pair of phrases, for each phrase we generate a vector that captures the *tf-idf* score of every lemma in context (the lemma frequency multiplied by the inverse document frequency). The context for this feature is delimited by 8 words before and after the phrase. The operational definition for a *document* for calculating the *idf* values is

[1] We are using Buckwalter transliteration for rendering Arabic script in ASCII [9].

therefore a segment of 16 words. We collected a relatively large set of such documents from Arabic Gigaword (4th ed.).

The vectors are relatively sparse, each containing merely 16 non-zero values at most, while their dimension is much larger. Therefore, to increase the influence of the contextual lemmas that co-occur with their corresponding phrase more often than by chance, we calculate the pointwise mutual information (PMI) of every lemma appearing in the context of a specific phrase by collecting occurrences from chunks of about 10M words extracted from Arabic Gigaword. Finally, the *tf-idf* value of each lemma is multiplied by the relevant PMI value. Accordingly, every phrase P is represented by the following vector:

$$V_P = \{\langle l, \text{tf-idf}(l, P) \times \text{PMI}(l, P) \rangle \mid l \in \text{context}(P)\}$$

Working with lemmas is natural. The lemma groups together all the inflected perfective and imperfective forms of a verb, and all the inflected singular, dual and plural forms of a noun. Contextual words that are either not derived from a lemma or for which the morphological analyzer failed to find the lemma are considered with their surface form instead. This situation happens mostly (but not only) with named entities; hence, each named entity occurring in the context of P is replaced by a placeholder representing the entity type (e.g. person, organization, location).[2] Then, given a pair of phrases, we measure the cosine similarity of their vectors, and use it as a single numeric feature for classification. We checked the distribution of the context-similarity score over a sample of 1,735 phrase pairs corresponding to 867 positive and 868 negative pairs. Figure 1 shows the distribution, where the abscissa represents subintervals of the context-similarity value ranging from 0 to 1, that is, the first column represents the values between [0, 0.08), the second column represents the values between [0.08, 0.16), and so on. As expected, we note that a greater mass of the negative pairs is concentrated at the lower end of the scale, while the positive pairs move toward the right-hand side. We conclude that the context-similarity feature cannot be used deterministically for deciding positive or negative cases, however it can be combined with more relevant features and potentially help in classification. In addition to the cosine similarity of the two phrases, we use the part-of-speech and base-phrase tags of each contextual word, up to 6 words before and after each of the two phrases, taking into account their relative position in the sentence (cf. [2,4]).

3.3 Evaluation of the Context Classifier

Overall we extracted about 12,000 phrase pairs of various lengths. In order to prefer precision over recall, the number of negative examples was selected to be twice as many as the positive examples. We ran a 10-fold cross validation; the precision was 84.7 and recall was 79 (F-measure was 81.7). Essentially, our classifier prefers precision to recall. It means that it is capable of distinguishing between contexts of identi-

[2] Named entities were found by AMIRA 2.0 [14].

cal phrases, on the lemma level, and the context of non-paraphrases. However, since the paraphrases we are looking for are not part of this training set, and, in fact, are not necessarily composed of identical phrases, we cannot estimate the performance on real data, based on these results.

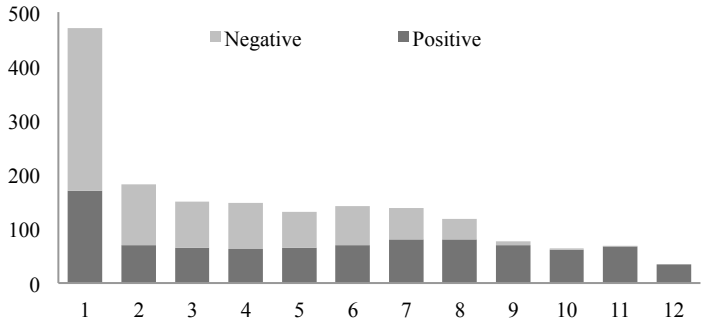

Fig. 1. The context-similarity distribution.

3.4 Deriving Paraphrases from a Monolingual Corpus

To extract paraphrases for a given phrase, we use 10M words of Arabic Gigaword as a resource, corresponding to 2.7M indexed phrases. However, although the size of the corpus may affect the coverage, it does not affect the quality of the results, as we do not use it to capture any statistical information (PMI values are calculated based on a different part of the corpus).

We preprocess the corpus with AMIRA 2.0 and extract phrases of up to 6 words, where phrases may not break a base phrase in the middle and must contain at least one content (non-functional) word. Every phrase was indexed in a database, so that it can be searched by each of its lemmas.

Given a phrase *P* for paraphrasing, our algorithm begins by searching the database for potential candidate paraphrases, defined heuristically as phrases that have at least some percentage of words in common with *P*, matched on the lemma level. Then it pairs each candidate phrase with *P*, and decides whether they are paraphrases or not using the context classifier. Theoretically, every phrase may be considered a potential paraphrase of *P*; however, checking every phrase from the database, given *P*, is computationally infeasible. For now, we consider phrases that have at least 40% of their lemmas in common, an ad-hoc threshold that was selected based on observations. A disadvantage of using this technique is that the extracted paraphrases are usually more structural. Considering better approaches, such as matching lemmas on the synonym level, is left for future investigation.

We consider the distribution provided by WEKA, which is calculated based on the distance of an instance from the separating hyperplane, to measure the quality of the returned paraphrases. In other words, we consider the distribution value as a confidence score. Moreover, to reflect the grammatical similarity of the phrases, that is,

whether the paraphrase may actually replace *P* without being detrimental to the input-sentence structure, we calculate two language-model scores. They are calculated on the text containing the paraphrase of *P* within the original context of *P*. One score is a language-model log-probability of the sequence of words that lie to the left of the paraphrase, including the first word of the paraphrase itself, and the second is calculated for the last word of the paraphrase followed by the sequence of words to the right of the paraphrase. Both scores measure how likely it is to find the paraphrase in the same context as *P*. The language model is generated using a large monolingual corpus, on the lemma level, with SRILM [37].

4 Using Paraphrases in Translation

We experiment with an Arabic-to-English implementation of Moses [26], a statistical-machine-translation platform, aiming to improve its translation quality using different levels of paraphrases of fragments of the input sentence. Paraphrases can be derived either for any fragment of the input text, or only for unseen phrases (regardless of the system's ability to translate them using the translations of [some of] their fragments.) Even if a phrase is in the phrase table, there is a chance that the overall translation may be improved by translating one of its paraphrases, due to wrong alignments resulting in a bad translation. Despite the hypothetical benefit of considering paraphrases of phrases that exist in the phrase table, there is a risk that the system will prefer a translation of one of the paraphrases that was incorrectly identified to the translation of the original phrase. To deal with this, we assign scores to the paraphrases that reflect the quality of their equivalence, so that the system will judge them accordingly.

We follow [24] and format every input sentence along with its paraphrases as a word lattice [18], that is, a directed acyclic graph, with every node uniquely labeled and every edge containing a token and a weight. A lattice is mainly used when parts of the input sentence are ambiguous and, instead of selecting merely one interpretation in the usual way, the lattice encodes multiple interpretations, each encoded with a plausibility weight.

Given a tokenized input Arabic sentence of *N* tokens for translation, we begin by initiating a lattice that captures the transition of the individual tokens linearly. (We use the D3 tokenization scheme [22].) We create a lattice of *N*+1 nodes and *N* edges, each representing a token. Every edge is assigned the value 1 (the maximum value in our case), keeping the lattice faithful to the input text. Then we add bypasses to the lattice, reflecting the paraphrases found by our paraphrasing algorithm. Paraphrases are generated for all phrases of the input sentence that are composed of up to 6 words and do not break a base phrase. To control the complexity we allow every phrase to have at most 3 paraphrases, each assigned with a confidence score higher than a threshold. The number 3 was determined mainly based on observations; the threshold is learned based on experiments, which we show in the next section. We refer to this structure as a *paraphrase lattice*. Figure 2 shows a paraphrase lattice representing the

input sentence *a b c d* augmented with one paraphrase of 4 tokens *x y z w*, replacing the phrase *b c*.

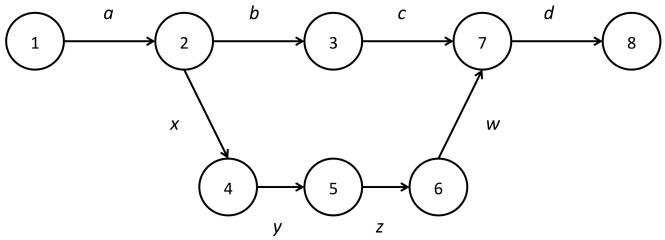

Fig. 2. Paraphrase lattice.

We assign weights to every edge in the lattice to reflect the chance that a specific paraphrase represents its corresponding input phrase. Those weights are considered by the decoder as part of the log-linear model of the translation system. In particular, Moses introduces an additional feature function, referred to as *InputFeature*, which represents the input type; the weight of that feature function, as combined in the log-linear model, allows the decoder to consider different paths of the input lattice while keeping in mind other factors, such as the translation and language models. The weight of InputFeature may be either set manually or tuned automatically with, for example, Minimum Error Rate Training (MERT) [31] on translated lattices. We take both approaches: (1) tuning the translation system on plain translated segments and setting the weight of the InputFeature function manually; and (2) tuning the translation system on translated lattices, and as a result, adapting the weights of all feature functions (including InputFeature) automatically.

Moses allows one to assign edges with several weights, whereas the log-linear model considers each individually. In this paper, we use three weights: (1) the context-classifier confidence score; (2) the left language-model score; and (3) the right language-model score. The weight of a single outbound edge is always 1. When there are several edges departing from the same node (where a paraphrase path begins), we normalize the score by dividing each component by the total sum of all values of the same component on the other sibling edges.

5 Experimental Approach and Results

We use Arabic paraphrases in translation and automatically measure their effect on the translation quality. Our baseline is an Arabic-to-English Moses instance, using different sizes of bilingual corpora, focusing on the newswire domain. We employ an English 5-gram language model, generated from a monolingual corpus of about 30 million words, and tune the system with MERT using bilingual texts containing 130K Arabic words. The Arabic text is tokenized following D3 [22] using MADA 3.1 [21,36] and the English text is tokenized using the default Moses tokenizer.

We test all the systems on the same evaluation set, the newswire part of the 2009 NIST OpenMT Evaluation set [20], containing 586 sentences, corresponding to 20,671 D3 tokens.

We begin by investigating how the threshold on the confidence score of the context classifier affects the overall translation performance. Clearly, as we decrease the threshold, the number of generated paraphrases grows larger; however, the quality of the paraphrases is likely to decrease (recall that we limit every phrase with maximum number of 3 paraphrases). The values of the confidence score that we observe are mainly concentrated in the range [0.91, 1]. Therefore, we use several threshold values within that range. Figure 3 shows the BLEU [32] scores calculated for a system that uses a bilingual corpus containing one million Arabic words, running under different threshold values. It is clear from the results that, when using a relatively high threshold, the translation quality gets better, while the number of generated paraphrases decreases. At 0.97 we see a steep drop in the number of qualified paraphrases, and as a result the BLEU score slightly declines. Overall, the results are encouraging, as we may learn from this that the confidence score affects the translation results as expected: With low thresholds, we get a relatively large number of paraphrases that do not appropriately reflect the meaning of its generating phrase, hence may be detrimental to the final result. Accordingly, we use 0.96 in the following experiments.

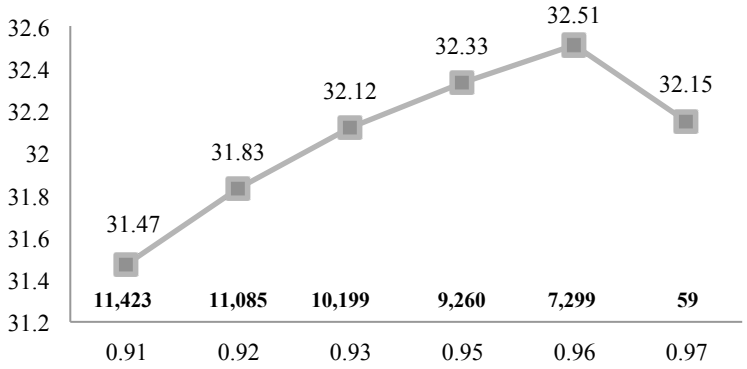

Fig. 3. BLEU scores of a system running on paraphrase lattice with different thresholds on the confidence score as returned by the context classifier. The abscissa represents the threshold values and the ordinate is the corresponding BLEU score. The numbers in boldface indicate the overall number of paraphrases generated using each threshold value.

To evaluate the contribution of paraphrasing to the translation results, we setup several baseline frameworks. Instead of deriving paraphrases from a corpus, we consider verbal and nominal synonyms, extracted with the help of a thesaurus. Since Arabic WordNet [7] is limited, we generated an Arabic thesaurus copying the simplistic technique of [3], namely, we look at the list of lemmas provided by SAMA 3.1 [28] and extract pairs of lemmas that share at least one English gloss in common.

Those pairs are deemed synonyms. Overall we extracted about 120K pairs corresponding to about 20K lemmas.

As for paraphrases, synonyms of input words are added to the lattice. Since synonyms are single words, we allow any number of synonyms to be generated for a single input word. As we do not have a confidence score under this setting, we assign equal weight to all synonyms, including the word itself. The language-model scores are calculated in the same way as for paraphrases. Note that synonyms are provided on the lemma level; hence, they must be inflected to reflect the form of the original input word. For example, given an input verb *EvrwA*, "they discovered", derived from the lemma *Eavar-u_1*, the thesaurus returns the synonym *ka$af-i_1*. To generate the required form *k$fwA*, we employ Almor [21], an Arabic morphological generator, and provide it with the morphological features of the original word, as extracted by MADA.

Given a verb, our paraphrasing algorithm very often identifies different inflected forms of the same verb as paraphrases. For the most part, such forms can be generated deterministically, regardless of the context in which they occur. Therefore, we build another baseline lattice, *Morph Gen*, which contains all the inflected forms generated by Almor that have the same English translation as the original form. For example, for a verb given in its perfective-form/singular/3rd-person, we generate only the form that is inflected for the opposite gender. This is because in English, singular 3rd person, past-tense verbs are combined with *s* or *es*. We manually crafted a few more such rules.

We experiment with different sizes of bilingual corpora and the BLEU scores are presented in Table 2. The best improvement over the baseline is 0.91 in BLEU score, observed when using a lattice containing paraphrases and all synonyms, running with a bilingual corpus of 500K Arabic words. When we increase the size of the bilingual corpus, the improvement is eroded, although it persists. This observation complies with the observations made in similar works [11,29]. The system that uses paraphrases outperforms all other baselines, including those that use synonyms. Moreover, we observe that synonyms moderately improve the final translations over the baseline when using a relatively small bilingual corpus. Synonyms also help further improve the results when combined with paraphrases. Using different inflected forms, represented by Morph Gen, was found to be counterproductive. In fact, the results obtained from using the Morph-Gen lattice are consistently a little worse than the baseline. This teaches us that the improvement obtained by using paraphrases was not due to verbs that get paraphrased simply as different inflected forms, (although such cases do exist). We used paired bootstrap resampling [25] for calculating statistical significance ($p < 0.05$) over the baseline. The improvements we get by paraphrases, syn/paraphrases on all corpus sizes are all statistically significant. The improvements we get by the synonym lattices are not significant, however.

Table 3 shows the total number of phrases for which the system generated at least one paraphrase/synonym, corresponding to the portion of them that do not appear in the bilingual corpora we use. Generally speaking, we learn that among the phrases that got paraphrased by the system, there are more phrases that appear in the bilingual corpora the translation system uses as a resource. It implies that the system benefits

from paraphrases of phrases that could be translated merely using the bilingual texts in the usual way.

Table 2. Evaluation results for different size (in millions of Arabic words) bilingual corpora on different lattices. Improvements over the baseline are in boldface.

Corpus size →	0.5M	1M	1.5M	4.5M
Baseline	31.48	32.18	32.75	34.20
Verb Synonyms	31.34	32.06	32.38	34.11
Noun Synonyms	**31.60**	**32.20**	32.26	33.97
All Synonyms	**31.50**	**32.31**	32.30	34.07
Morph Gen	31.44	32.00	32.40	34.02
Paraphrases	**32.28**	**32.51**	**33.19**	**34.21**
Syn/paraphrases	**32.39**	**32.72**	**33.28**	34.12

Table 3. The number of unseen phrases that were paraphrased by the system. The *Total Generated* column represents the total number of phrases for which the system created at least one paraphrase. Every other column represents the number of unseen phrases corresponding to the size (in millions of Arabic words) of the bilingual corpus used by the translation system.

	Total Generated	Unseen in 0.5M	Unseen in 1M	Unseen in 1.5M	Unseen in 4.5M
Verb Synonyms	466	34	32	32	29
Noun Synonyms	649	25	22	18	14
All Synonyms	1,115	59	54	50	43
Paraphrases	7,299	331	217	211	193
Syn/paraphrases	8,414	390	225	219	199

Table 4 compares the way some seen/unseen phrases got translated by the system with and without paraphrasing.

So far, all our experiments were executed on a system that was merely tuned on the original sentences, formatted as word lattices, but including neither paraphrases nor synonyms. The weight of the InputFeature function, which affects the preferences of the decoder, was assigned arbitrarily to be 0.1. As a next step, we repeat the same experiments, minus the less productive ones, this time with a system that was tuned with MERT on the same development set, formatted as paraphrase lattices and augmented with paraphrases. The results are presented in Table 5. The best statistically significant improvement of +1.73 (5.49%) BLEU points over the baseline is obtained by the system that uses 500K Arabic words, tuned with MERT on the paraphrase lattices. For the most part, tuning the parameters for paraphrases helps improve the

translations. But we see a slight drop for the larger corpus, suggesting that the weights assigned to other features were slightly miscalculated. We are in the process of trying to alleviate this.

Table 4. Examples of paraphrases and their translations. The Arabic text is tokenized according to the D3 scheme [22]. Columns from left to right: (1) the original Arabic phrase; (2) the way it was translated by the baseline system (<unseen> means that the phrase was not translated as a whole); (3) the paraphrase that was used by a system with paraphrasing capabilities; (4) the way the paraphrase was translated; and (5) our comments.

Original phrase	Baseline translation	Paraphrase	Paraphrase Translation	Comments
dblwmAsywn bwlndywn	<unseen>	*dblwmAs bwlndy*	Polish diplomat	Different number
Al+ qyAdp Al+ Eskryp Al+ jnwbyp	<unseen>	*Al+ qyAdp Al+ Eskryp fy Al+ jnwb*	the military headquarter of the south	Different phrasing
w+ y}n Al+ Tfl	the Child and *y}n*	*lmA*A bkY Al+ Tfl*	why the child cried	Synonyms helped to improve translation
nATq b+ Asm Al+ xArjyp	a spokesman of the foreign affairs	*Al+ mtHdv b+ Asm wzArp Al+ xArjyp*	the spokesman of the ministry for foreign affairs	
wADAf >n AlEskryyn	The military *wADAf*	*ADAf >n Aljy$*	he added that the army	Wrong tokenization (*wADAf*) was fixed through paraphrasing
Al+ A$hr Al+ Axyrp	the last months	*Al+ AyAm Al+ Axyrp*	the last days	Wrong paraphrasing resulting in wrong translation
nzE Al+ slAH Al+ nwwy	The elimination of weapon for mass destruction	*Ant$Ar Al+ AslHp Al+ nwwyp*	the spreading of weapon for mass destruction	Antonyms are identified wrongly as paraphrases

6 Conclusions

We have demonstrated the potential of using Arabic paraphrases and synonyms to improve the results of a statistical translation system. We presented a new technique for paraphrasing from a monolingual corpus, supported by a context classifier that was trained using examples from a relatively small set of comparable documents. As

a result, the resulting algorithm does not require large quantities of text to calculate word statistics. Arabic is highly inflected; therefore, working on the lemma level was natural. Although some of the derived paraphrases were in fact different inflected forms of their corresponding original phrases, we found that this was not the salient reason for improvement. We configured our algorithm to prefer precision over recall by merely considering phrases that have some lemmas in common with the subject phrase. Improving this technique should improve the results even more. We may conclude that the translation system benefits from using MERT on paraphrase lattices to adjust the weight of the InputFeature function, resulting in better final translations. Our immediate intent is to apply this paraphrasing technique to additional languages with complex morphology.

Table 5. Results of some of the experiments from Table 2, repeated after tuning the system with paraphrases. (TOPL = tuned on paraphrase lattice) indicates that the system was tuned on paraphrase lattices.

Corpus size →	0.5M	1M	1.5M	4.5M
Baseline	31.48	32.18	32.75	34.20
All Synonyms	31.50	32.31	32.30	34.07
All Synonyms (TOPL)	**31.89**	**32.47**	**32.45**	33.73
Paraphrases	32.28	32.51	33.19	34.21
Paraphrases (TOPL)	**33.01**	**33.11**	**33.46**	34.19
Syn/paraphrases	32.39	32.72	33.28	34.12
Syn/paraphrases (TOPL)	**33.21**	**33.43**	**33.68**	34.10

References

1. Bannard, C., Callison-Burch C.: Paraphrasing with Bilingual Parallel Corpora. In: The 43rd Meeting of the Association for Computational Linguistics (ACL), pp. 597-604. Ann Arbor, MI. (2005)
2. Bar, K., Dershowitz, N.: Deriving Paraphrases for Highly Inflected Languages from Comparable Documents. In: The 24th International Conference on Computational Linguistics (COLING). Mumbai, India (2012)
3. Bar, K., Dershowitz, N.: Using Semantic Equivalents for Arabic-to-English Example-Based Translation. In: Soudi, A., Farghaly, A., Neumann, G., Zbib R. (eds.) Challenges for Arabic Machine Translation, pp. 49-72. John Benjamins Publishing Company (2012)
4. Barzilay, R., McKeown, K.: Extracting Paraphrases from a Parallel Corpus. In: The 43rd Meeting of the Association for Computational Linguistics (ACL), pp. 50-57. Toulouse, France (2001)

5. Barzilay, R., Lee, L.: Learning to Paraphrase: An Unsupervised Approach Using Multiple-Sequence Alignment. In: The North American Chapter of the Association for Computational Linguistics (HLT-NAACL), pp. 16-23. Edmonton, Canada (2003)
6. Benajiba, Y., Diab, M., Rosso, P.: Arabic Named Entity Recognition: An SVM-based Approach. In: The Arab International Conference on Information Technology (ACIT). Hammamet, Tunisia (2008)
7. Black, W., Elkateb, S., Vossen, P.: Introducing the Arabic WordNet Project. In: The 3rd Global Wordnet Conference (GWC), pp. 295-299. Jeju, South Korea (2006)
8. Blum, A., Mitchell, T.: Combining Labeled and Unlabeled Data with Co-training. In: The 11th Annual Conference on Computational Learning Theory (COLT), pp. 92–100. Madison, WI (1998)
9. Buckwalter, T.: Buckwalter Arabic Morphological Analyzer Version 1.0. LDC Catalog number LDC2002L49 (2002)
10. Callison-Burch, A.: Syntactic Constraints on Paraphrases Extracted from Parallel Corpora. In: The Conference for Empirical Methods in Natural Language Processing (EMNLP), pp. 196-205. Honolulu, Hawaii (2008)
11. Callison-Burch, C., Koehn, P., Osborne, M.: Improved Statistical Machine Translation using Paraphrases. In: The North American Association for Computational Linguistics (NAACL). New York City, NY (2006)
12. Chang, C. C., Lin, C. J.: LIBSVM: A Library for Support Vector Machines. ACM Transactions on Intelligent Systems and Technology, 2:27:1-27:27 (2011)
13. Dagan, I., Glickman, O.: Probabilistic Textual Entailment: Generic Applied Modeling of Language Variability. In: PASCAL workshop on Learning Methods for Text Understanding and Mining, pp. 26-29. Grenoble, France (2004)
14. Diab, M.: Second Generation Tools (AMIRA 2.0): Fast and Robust Tokenization, POS Tagging, and Base-Phrase Chunking. In: MEDAR 2nd International Conference on Arabic Language Resources and Tools. Cairo, Egypt (2009)
15. Dolan, W. B., Quirk, C., Brockett, C.: Unsupervised Construction of Large Paraphrase Corpora: Exploiting Massively Parallel News Sources. In: The 20th International Conference on Computational Linguistics (COLING). Geneva, Switzerland (2004)
16. Duboue, P. A., Chu-Carroll, J.: Answering the Question you wish They had Asked: The impact of Paraphrasing for Question Answering. In: Human Language Technology-North American Association for Computational Linguistics (HLT-NAACL), pp. 33-36. New York City, NY (2006)
17. Dunning, T.: Accurate Methods for the Statistics of Surprise and Coincidence. Computational Linguistics 19, 1, 61-74 (1993)
18. Dyer, C., Muresan, S., Resnik, P.: Generalizing Word Lattice Translation. In: The 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT), pp. 1012–1020. Columbus, Ohio (2008)
19. Fader, A., Zettlemoyer, L., Etzioni, O.: Paraphrase-Driven Learning for Open Question Answering. In: The 51st Meeting of the Association for Computational Linguistics (ACL), Sofia, Bulgaria (2013)
20. Garofolo, J.: NIST OpenMT Eval 2009, http://www.itl.nist. gov/iad/mig/tests/mt/2009/.
21. Habash, N. Rambow, O.: Arabic Tokenization, Morphological Analysis, and Part-of-speech Tagging in One Fell Swoop. In: The Conference of American Association for Computational Linguistics, pp. 578-580. Ann Arbor, MI (2005)
22. Habash, N., Rambow, O., Roth, R.: MADA+TOKAN: A Toolkit for Arabic Tokenization, Diacritization, Morphological Disambiguation, POS Tagging, Stemming and Lemmatiza-

tion. In: The Second International Conference on Arabic Language Resources and Tools, The MEDAR Consortium. Cairo, Egypt (2009)
23. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I. H.: The WEKA Data Mining Software: An Update. SIGKDD Explorations, vol. 11, pp. 10-18 (2009)
24. Jinhua, D., Jiang, J., Way, A.: Facilitating Translation Using Source Language Paraphrase Lattices. In: The Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 420-429. Stroudsburg, PA (2010)
25. Koehn, P.: Statistical Significance Tests For Machine Translation Evaluation. In: The Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 388-395. Barcelona, Spain (2004)
26. Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., Herbst, E.: Moses: Open Source Toolkit for Statistical Machine Translation. In: The 45th Meeting of the Association for Computational Linguistics (ACL). Prague, Czech Republic (2007)
27. Lin, D., Pantel, P.: Discovery of Inference Rules for Question Answering. Natural Language Engineering 7(4), 343-36 (2001)
28. Maamouri, M., Graff, D., Bouziri, B., Krouna, S., Bies, A., Kulick, S.: Standard Arabic morphological analyzer (SAMA), Version 3.1. Linguistic Data Consortium, Philadelphia, PA (2010)
29. Marton, Y., Callison-Burch, C., Resnik, P.: Improved Statistical Machine Translation Using Monolingually-Derived Paraphrases. In: The Conference on Empirical Methods in Natural Language Processing (EMNLP). Singapore (2009)
30. Nakov, P., Ng, H. T.: Translating from Morphologically Complex Languages: A Paraphrase-Based Approach. In: The Meeting of the Association for Computational Linguistics (ACL). Portland, OR (2011)
31. Och, F. J.: Minimum Error Rate Training in Statistical Machine Translation. In: The 41st Annual Meeting of the Association for Computational Linguistics (ACL), pp. 160-167. Sapporo, Japan (2003)
32. Papineni, K., Roukos, S., Ward, T., Zhu, W. J.: Bleu: A Method for Automatic Evaluation of Machine Translation. In: The 40th Annual Meeting of the Association for Computational Linguistics (ACL), pp. 311-318. Philadelphia, PA (2002)
33. Parker, R., Graff, D., Chen, K., Kong, J., Maeda, K.: Arabic Gigaword, Fourth Edition. Linguistic Data Consortium, LDC2009T30, ISBN 1-58563-532-4, Philadelphia, PA (2011)
34. Quirk, C., Brockett, C., Dolan, W.: Monolingual Machine Translation for Paraphrase Generation. In: The 2004 Conference of Empirical Methods in Natural Language Processing (EMNLP), pp. 142-149. Barcelona, Spain (2004)
35. Riezler, S., Vasserman, A., Tsochantaridis, I., Mittal, V., Liu, Y.: Statistical Machine Translation for Query Expansion in Answer Retrieval. In: The 45th Annual Meeting of the Association for Computational Linguistics (ACL), pp. 464-471. Prague, Czech Republic (2007)
36. Roth, R., Rambow, O., Habash, N., Diab, M., Rudin, C.: Arabic Morphological Tagging, Diacritization, and Lemmatization Using Lexeme Models and Feature Ranking. In: The 46th Annual Meeting of the Association for Computational Linguistics (ACL), pp. 117-120. Columbus, Ohio (2008)
37. Stolcke, A.: SRILM -- An Extensible Language Modeling Toolkit. In: The International Conference on Spoken Language Processing, pp. 901-904. Denver, CO (2002)
38. Vapnik, V., Cortes, C.: Support Vector Networks. Machine Learning, vol. 20, pp. 273-297 (1995)

39. Wang, R., Callison-Burch, C.: Paraphrase Fragment Extraction from Monolingual Comparable Corpora. In: The Fourth Workshop on Building and Using Comparable Corpora (BUCC). Istanbul, Turkey (2011)
40. Zhao, S., Wang, H., Liu, T., Li, S.: Pivot Approach for Extracting Paraphrase Patterns from Bilingual Corpora. In: The 46th Annual Meeting of the Association for Computational Linguistics (ACL), pp. 780-788. Columbus, OH (2008)