IJICTR International Journal of Information & Communication Technology Research

Volume 5- Number 4- Autumn 2013 (47-58)

Using Synchronous TAG for Source-Side Reordering in SMT

Amin Mansouri
School of Electrical and Computer Engineering,College of Engineering,
University of Tehran,
Tehran, Iran
a.mansouri@ece.ut.ac.ir

HakimehFadaei
School of Electrical and Computer Engineering,College of Engineering,
University of Tehran,
Tehran, Iran
h.fadaei@ece.ut.ac.ir

HeshaamFaili
School of Electrical and Computer Engineering,College of Engineering,
University of Tehran,
Tehran, Iran
hfaili@ut.ac.ir

Mohsen Arabsorkhi
School of Electrical and Computer Engineering,College of Engineering,
University of Tehran,
Tehran, Iran
marabsorkhi@ece.ut.ac.ir

Received: November14, 2012- Accepted: August 8, 2013

Abstract—**Recent efforts in machine translation try to enrich statistical methods by syntactic information of source and target languages. In this paper we present a hybrid machine translator, which combines rule-based and statistical models in a serial manner. This system uses synchronous tree adjoining grammar (STAG) to benefit the context sensitivity of this formalism. In this system, a set of reordering rules in STAG formalism is automatically extracted from a parallel corpus. These rules are used to change the word orders of the source sentence to match the word ordersin the target language. The restructured sentences are then translated to target language using a statistical approach. Experiments are carried out on three different datasets for English-Persian translation. Experimental results show that the presented reordering method combined with conventional or monotone phrase-based SMT, improves the translation quality respectively by 1.8 and 0.55 points regarding BLEU score.**

Keywords-Statistical Machine Translation, Reordering Rules, Tree Adjoining Grammar

I. INTRODUCTION

Statistical machine translation (SMT) concerns the translation of a source text to a target language using statistical models. The parameters of this statistical model are learned automatically using a parallel corpus. This paradigm follows a Bayesian noisy channel model, which tries to find the most probable target language sentence for the source sentence. The probability is calculated based on two models: Translation and language models. The former is responsible for the faithfulness and the latter for the fluency of translation.

In recent years, many variations of SMT have been proposed, each of which tries to improve the quality of translation. In this trend we start from conventional SMT systems which use language and translation models with the goal of maximizing the probability of their translated sentences [1]. These systems are corpus-based in which all the model parameters are learned automatically form parallel corpora and there is no need for manual efforts.

Conventional SMT bridges from word-based models, which ignore contextual information, to phrase-based models by extracting the phrase translations from word based models [16]. In phrase-based SMT the reordering and the word sense disambiguation problems are solved by considering phrases (word sequences) as the building block in translation. Pure SMT systems merely rely on word sequences and totally ignore linguistic concepts such

as constituents and grammar rules. In other words the huge amount of syntactic knowledge, embedded in sentences is simply put aside in SMT systems. This may work for structurally close languages but what if we are dealing with two languages with totally different structures. How far can conventional SMT models tolerate this divergence in structure?

Languages may be very different from the structural point of view. Word order is a major issue in translation. Many re-orderings would be needed to transform the structure of a sentence in one language to its corresponding structure in another language. In the large scale we should handle translation between SOV and SVO languages. Re-orderings may appear in smaller scales like adjective-noun ordering. Phrase based SMT systems can handle short distance re-orderings by their phrase definition but, when it comes to long distance re-orderings, as the distance may fall beyond the phrase size, these systems lose their accuracy.

To deal with the reordering problem some systems change the orderings of the source side text in a preprocessing step and then use a SMT model to do the translation. Earlier systems used POS information of the sources side text to decide about necessary reorderings[6][7]. Further studies showed that syntactic information could be of more help and that is when hybrid systems combined syntax-based MT and statistical MT. Syntax-based SMT has different variations such as string to tree, tree to string and tree to tree models. In these models the syntactic information of the source-side or target-side or both is used in translation.

In tree to string [2][30]approach a parse tree of the source sentence is given as input. The system feeds this tree to the noisy channel to get a sentence in target language. It means that these systems use the syntactic knowledge of the source language in translation while in string to tree approach[8][3]the case is reverse and syntactic model of the target language in needed. In tree to tree models, we have the parse trees of both sides. These approaches require efficient parsers, which are not available for all languages.

The systems that use syntactic information in the process of translation may use different grammar formalisms. Many systems use CFGs for parsing ([4][5]) but, as mentioned in [8]this formalism has some shortcomings which have encouraged the researchers to search an alternative for CFG. During recent years, Tree Adjoining Grammars have received much attention as an alternative formalism for CFG[20]. TAGs are more powerful than CFGs in Chomsky hierarchy as they are considered as mildly context-sensitive grammars. In TAG elementary trees are the building blocks and in each operation we replace a node with a tree. TAG has a larger domain of locality than CFG which makes it more powerful in detecting arguments of different predicates throughout the sentence[12]. In the task we are addressing in this paper we need synchronous grammars. Synchronous grammar is a generalization over grammar definition, which relates two languages. With synchronous grammars two sentences in two languages could be generated with the same structure. These grammars are widely used in machine translation. For the reasons mentioned above, we have used synchronous TAG (STAG) [10] as our formalism.

In this paper we present a SMT model which uses syntactic information of the source side sentences to extract reordering rules from a word-aligned parallel corpus. These rules are then used in the pre-processing step of translation to change the word orders of the sources sentence to match the orderings in the target language. In our model the source-side sentences are parsed to form derivation trees. Unlike some systems such as [8] and [9] which use STAG in their decoding process, we use it in our pre-processing step and apply our extracted rules to the derivation trees of the source sentence. This model also uses word classes to generalize the extracted rules for unseen words. The extracted rules can be utilized in other SMT or rule-based systems as well.

We used our model in English to Persian translation. Persian is an Indo-European language which is spoken in countries such as Iran, Afghanistan and Tajikistan. Persian is categorized as a highly inflectional and free-word-order language. The sentences in Persian follow a SOV structure while English is SVO. The free-word -order characteristic of Persian make translation to and from this language more difficult. It's because extracting reordering rules that can model phrasal movements is more challenging. Another difficulty in translating to Persian is the morphologically rich nature of this language. For example, in Persian verbs could contain a series of postfixes indicating tense, person and object, which does not exist in English. Detecting the right tense and person from English sentence and reflecting them in Persian translation, is one the challenges that statistical approaches face. This characteristic of Persian leads to generation of many different word forms in a corpus. Since all the word forms are not frequent, in many cases, this diversity of forms causes sparseness. Therefore to have a rich and accurate translation model, large parallel corpora are needed which are not available in Persian.

The proposed model is described in detail throughout the paper which is organized as follows: Section 2 discusses the previous work done in this domain. In Section 3 we present an overview of our system and its architecture. The complete definition of reordering rules and their extraction process are explained in Section 4. Finally the experiments performed by using our model and the achieved results will be discussed in Section 5.

II. Previous Work

Phrase-based SMT systems consider sequences of words as their building block. In these systems the linguistic notion of "phrase" does not exist and any sequence of words in the input data could be considered as a phrase. This fact limits the generalization abilities of these systems, which will highly affect the accuracy of translation to/from free word order languages, while in syntax-based MT we deal with general linguistic rules which can be applied more properly on input sentences. To benefit the advantages of both models, researchers enrich SMT

models with syntactic information. This information could either be used during the translation or in pre-processing. In these systems rules are extracted automatically using statistical approaches. This section reviews some related systems that extract reordering rules and combine them with statistical approach in translation. It should be mentioned that reordering rules are not limited to syntactic rules.

Some systems use POS tags as their guide to reordering. In these works a set of reordering rules are extracted from a parallel corpus annotated by POS tags. The rules may merely involve POS tags or may be on a combination of words and POS tags. In [17] a reordering method is proposed to change the word orders of the source sentence. In this work a set of reordering rules, concerning POS tags and words, are extracted from a parallel corpus. In the translation process, first a word lattice is generated using all potential reorderings on the original sentence. Then by considering the probabilities of the exploited rules the most probable set of reordering is chosen and performed on the source side sentence. The reordered sentence is finally translated in a conventional decoding step.

In other category of systems on which we focus in this section, syntactic rules are used to handle the reordering issue and they are applied on parse trees. The idea of using syntactic rules for reordering is used in many systems but the grammar formalisms vary.

Galley et al. [4]claimed that the syntactic transformation rules that can model the differences in word orders of parallel sentences, go far beyond simple one-level parent-child sub-trees. They propose a language-independent method for extracting syntactic transformation rules (which we call "reordering rules" in this paper) from an aligned parallel corpus and parse trees. This method takes parse trees in one language and corresponding sentence in another language with their alignment as input data. Given a source sentence, this method generates all possible derivations and creates an alignment graph for every valid derivation. The alignment graph is finally used to extract reordering rules. In [4], CFG is used to parse the source sentence. The extracted rules are evaluated in a string-to-tree SMT system. Deneefe et al. [8] use the same basic idea but address the main problem of the above method which lacks generalizability. To solve this problem they suggest using binarized trees and they use synchronous TIG in their tree-to-tree model. Using a parallel corpus as their train data, they extract some syntactic rules which involve tag operations and their probabilities. This system extracts four types of lexical transfer rules: rules involving substitution, rules involving adjunction and rules with multiple substitution and adjunction. These probabilistic rules are finally used in the decoder to generate the target tree from the source sentence's tree.

Among other systems which use TAG as their grammar formalism we can mention [9] which extracts a set of tree to string reordering rules and uses them in its decoding step to reach the target sentence. This work has much in common with [4] and[8]. In this system, the decoding process itself is performed in a tree to string manner. This system extract three types of rules and by combining them achieves a set of composed rules. These composed rules are generated on the fly in decoding phase to avoid adjoining. As a result the decoding merely face substitution rules.

Another structure that can be of great help in reordering is dependency structure. In some systems like[23], dependency structures are used to handle long-distance reorderings. These structures can bring far apart dependencies closer and as a result, detecting and performing the reorderings is facilitated. Bach[23] proposes a reordering model based on source side dependency sub-tree movements. Two sub-tree movements are defined in this model: *inside* and *outside* which concerns moving a sub-tree inside or outside its source sub-tree. This model also takes into account the orientation of each phrase with regard to its previous one, which can be monotone, swap or discontinuous, by considering the alignments in train data. The proposed model achieved improvements in translating English to Spanish and English to Iraqi. Combinatory Categorial Grammar (CCG) is, as well, used in some works to provide syntactic information for statistical approaches [24][25]**Error! Reference source not found.**. The ability of CCG in handling long-range dependencies makes it suitable for the reordering task.

Xia and McCord [5] also propose an algorithm for automatic extraction of reordering rules, but unlike the above systems they use their extracted rules in the pre-processing step. In their work they use CFG rules to parse their source side sentences of a parallel corpus. After parsing the input sentences and aligning them with their corresponding sentence in target language, some lexicalized and unlexicalized reordering patterns are statistically extracted from input data. These rules are subsequently used in the translation process to transform the word orders of the source sentence to match the ordering of the target language. The newly generated source sentences are then translated to the target language using a conventional statistical translator. Collins [26] uses a similar method for source side reordering in pre-processing. But in this work a set of hand crafted rules are exploited. Collins et al.'s experimental results show an overall improvement in translation from German to English. Howlettand Dras[27] re-implement Collins' method in a dual-path SMT model, in which the original and the reordered sentence are provided to the translator in the form of a lattice. Their experiments show that, contrary to what was concluded in Collins et al.[26], reordering in pre-processing, if used alone, wouldn't improve the translation quality.Among other works that follow this trend we can mention the work of Habash[28] for Arabic to English and Wang et al. [29]for Chinese to English translation.

In our systems we follow the idea of using reordering rules in the pre-processing step but to alleviate the shortcomings of CFG we use synchronous TAG as our formalism. In this paper the reordering rules are extracted automatically from training data.

III. System Overview

The system presented in this paper exploits a serial hybrid approach in translation. In this system, SMT is enriched by reordering rules extracted from a parallel corpus. This system consists of two main modules: The rule-based reordering module and the SMT module. This section explains the overall architecture of the system and its main modules.

In our system a set of reordering rules is applied in the rule-based phase to prepare the text in source language to be used as an input for SMT module. These rules could be crafted manually, but the extraction process is time-consuming and that is why automatic methods could be of great help. In its starting step, our system uses an automatic method to extract reordering rules from a given parallel corpus. The extracted rules are refined and the noisy ones are eliminated from the final set. The extraction and refinement algorithms will be explained in details in Section 4.

When the reordering rules are determined, they are used to regenerate the source language corpus, with the word orders of the target language. Fig. 1 shows the application of some reordering rules on a sample sentence in English.

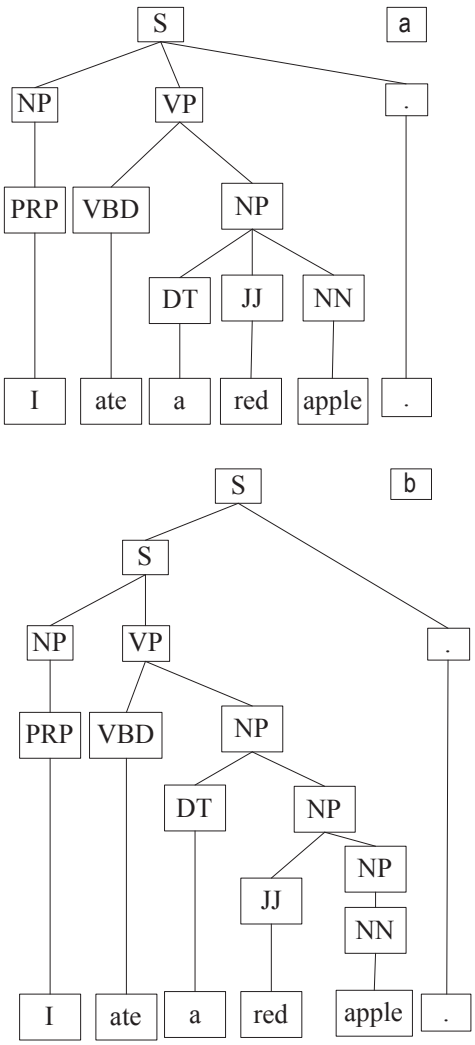

Figure 1. Reordering of the sentence "I ate a red apple" a) Parse tree of the original sentence b) Parse tree of the reordered sentence

As shown in Fig. 1, in this example two reordering rules are used, one of them reorders NP and VBD while the other swaps NN and JJ. These rules change our primary sentence "I ate a red apple." to "Ia apple red ate." in new corpus. This word ordering is compatible with target language word ordering, which is Persian in this example. As a result of the reordering process, when the sentences of the newly generated corpus are aligned with their corresponding sentence in the target corpus, we can find fewer word alignments that cross each other.

In the final step, the new parallel corpus in used as training data in our basic phrase based statistical machine translator. The phrase table is generated for this parallel corpus and the probabilities are learned.

After the training step, given a new sentence, the rule-based reordering module, reorders the source sentence with the extracted rules and finally the SMT module translates the reordered sentence.

IV. Reordering Model

As it was mentioned earlier our system uses STAG as the reordering rule formalism. In this section we briefly review STAG and describe how to generate derivation and derived trees. We used the same method as [12] to generate these trees. The rest of the section explains the rule extraction and refinement process.

A. Synchronous Tree Adjoining Grammar

Tree adjoining grammar (TAG) was first introduced in 1975 by Joshi *et al.* [11]. This formalism is more powerful in describing the characteristic of natural languages as it is considered as a mildly context sensitive formalism.

The building blocks of TAG are called *Elementary Trees*, which are anchored by at least a lexical term. Elementary trees are combined using *substitution* and *adjunction* operations. These two operations are described in Fig. 2a. In parsing a sentence by TAG, we combine different elementary trees anchored by the terms in our sentence, to build the parse tree. The generated parse tree is called *derived tree* and the combination process is saved in another tree which is called *derivation tree*. Each node in the derivation tree involves an elementary tree. Fig. 2 [12] shows how a derived tree is generated for the sentence "underwriters still draft policies".

To generate derivation and derived trees, the input sentence should be parsed. The parsing process includes two steps: pre-parsing with PCFG formalism and generating derivation trees. Given an input sentence, our system uses Stanford Parser [18] to obtain a CFG parse tree. The resulted CFG parse tree is then converted to a derivation tree. This process is accomplished in three steps: first the derived tree is generated from the parse tree as it is described in [12]. Then the derived tree is decomposed to its constructing elementary trees. The history of this decomposition is meanwhile stored in the derivation tree.

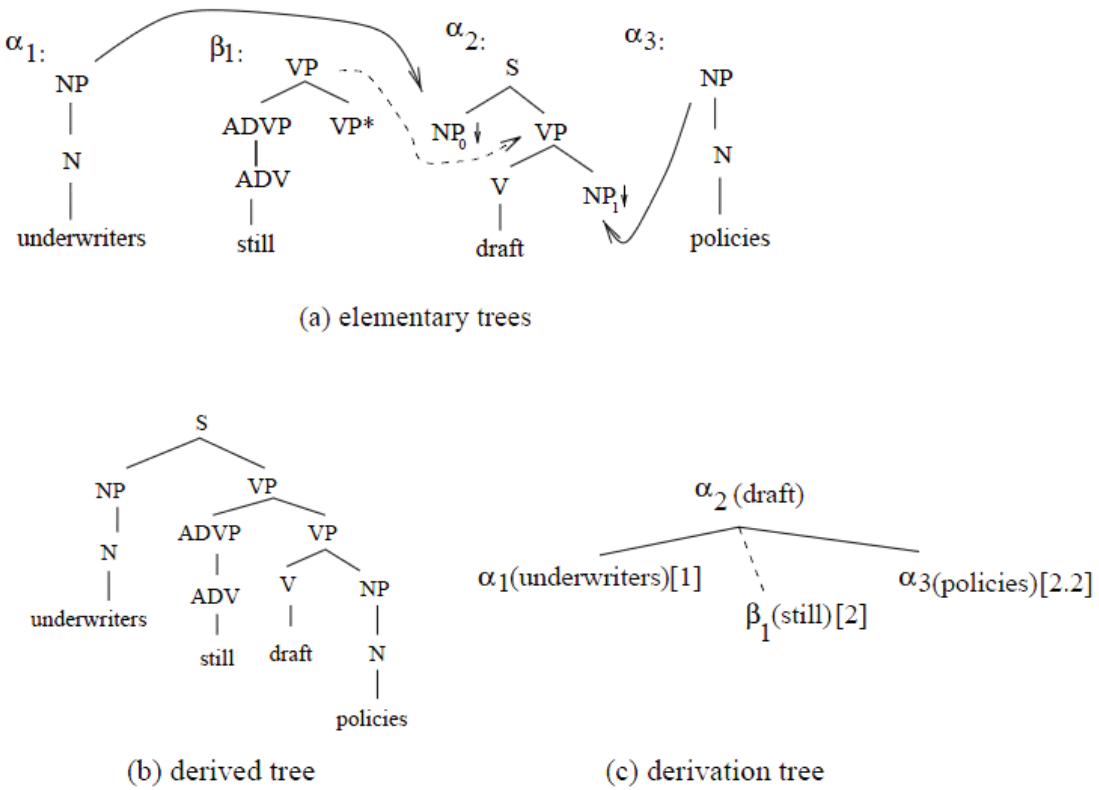

Figure 2. Elementary, derived and derivation trees for the sentence "underwriters still draft policies" [12]

As mentioned earlier, in this system, we need a synchronous grammar. Thus synchronous TAG (STAG) is selected as our formalism. In STAG we deal with tree pairs in source and target languages. In each pair corresponding nodes are related to each other in the sense that if we perform an adjunction or a substitution operation on a node in the source language's tree, the same operation would be done on the related node in the target language's tree. Two examples of synchronous TAG, with and without reordering, are depicted in Fig. 3.

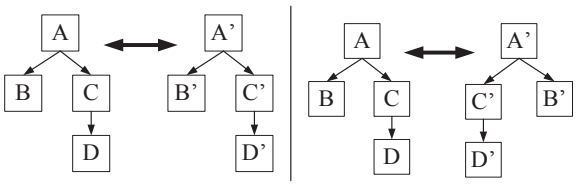

Figure 3. STAG – left side without reordering and right side with reordering

B. Reordering-Rule Extraction

The rule extraction module gets the following data as input:

- A word aligned parallel corpus: The input corpus is sentence aligned and the word alignments are extracted using GIZA++ toolkit [19].To the best of our knowledge GIZA++ is the most widely used alignment toolkit in the field of SMT and has been used in many recent works [9][22]. This tool uses and arbitrary combination of IBM and Hidden Markov alignment models to perform word level alignment.
- Derivation and derived trees: the sentences of the source side in parallel corpus are parsed by Stanford parser. The LexTract algorithm is then applied on the parse trees and the derivation and derived trees are generated.

Utilizing the above data, our system extracts and refines two types of reordering rules: lexicalized and unlexicalized rules. In the rest of the extraction process, elementary trees are considered as the input of the algorithm.

It is worth mentioning that not all of the generated trees are suitable for rule extraction in this system. Our system applies some constraints to choose suitable elementary trees from the input set. The valid elementary trees have the following characteristics:

- None of their anchors are aligned to null.
- In case of having one-to-many alignments, the corresponding target words are sequential. It means, if a node *x* in the elementary tree is anchored with a sequence of n words in the target sentence starting from w_i and ending to w_j (where i<j), any w_k (i<k<j) which is not anchored to *x* should be aligned to null.
- They do not have many-to-one alignments. It means if any of the anchored words is aligned to other words in the source sentence, the elementary tree is marked as invalid.

Thus to start the extraction process we first exclude the invalid elementary trees from the input set. Our final goal is to reorder the source sentence in such a way that the result sentence has no crossing alignments with the target sentence. In order to ensure the existence of such reorderings, the mentioned constraints should be satisfied.

For each elementary tree in the input set and regarding its alignments with the target sentence, the algorithm generates an equivalent tree in such a way that no crossing alignment remains. As an example we again use the sentence "I ate a red apple". Fig. 4 shows an elementary tree which is used in the parse of this sentence. As it can be seen in this example the original elementary tree (on the left) has a crossing alignment. The word "ate" in the source sentence is aligned to the w_4 in the target sentence while the rightmost NP is aligned to the word sequence w_1 w_2 w_3. The algorithm generates the tree shown in rightmost side of the figure, in which no alignments cross each other. Although this tree is still in the source language but, it matches the word orders of the target language.

In this case VBD and its anchor "ate" should be reordered with NP. This reordering can be considered as a candidate lexicalized rule. Another extractable lexicalized rule from this example sentence is the reordering of JJ and its anchor "red" with NP.

Table I summarizes all of the reordering rules that could be extracted from our example sentence. The leftmost column shows the elementary tree in English and the next column shows its reordered version in target language Persian. The anchors both in English and Persian are mentioned in next two columns, From now on we show trees by their pre-order traversal and adjunction and substitution are respectively shown by '*' and '+'. '@' is considered as a separator.

By applying these reordering rules to the elementary trees of the source language, new elementary trees with the ordering of the target language are generated. Then following the information recorded in the derivation tree, we can combine these new elementary trees to achieve a complete derived tree suitable for target language. The corresponding tree for our example sentence is pictured in Fig. 5. As shown, this tree has no crossing alignments with the target sentence.

Some elementary trees may stay unchanged during this process, e.g. the row with bold text in Table I. The number of such trees is higher in structurally close languages while for structurally divergent languages this number is much lower.

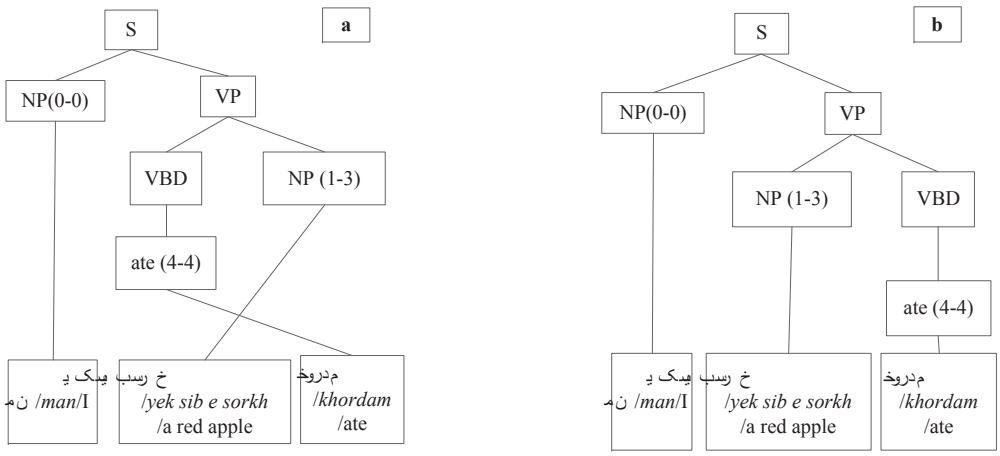

Figure 4. a: The elementary tree before reordering b: The elementary tree after reordering which reflects the word order of the target

TABLE I. SOME EXTRACTED STAG RULES FROM THE SAMPLE SENTENCE "I ATE A RED APPLE"

Elementary tree in English	Reordered elementary tree (Persian word orders)	Anchor	Persian Anchor
(S (NP+) (VP (VBD) (NP+)))	(S (NP+) (VP (NP+) (VBD)))	ate	خوردم/*khordam*
(S (S*) (. .))	(S (S*) (. .))	.	.
(NP (PRP))	(NP (PRP))	I	من /*man*
(NP (JJ) (NP*))	(NP (NP*) (JJ))	red	سرخ/*sorkh*
(NP (DT) (NP*))	(NP (DT) (NP*))	a	یک/*yek*
(NP (NN))	(NP (NN))	apple	سیب/*sib*

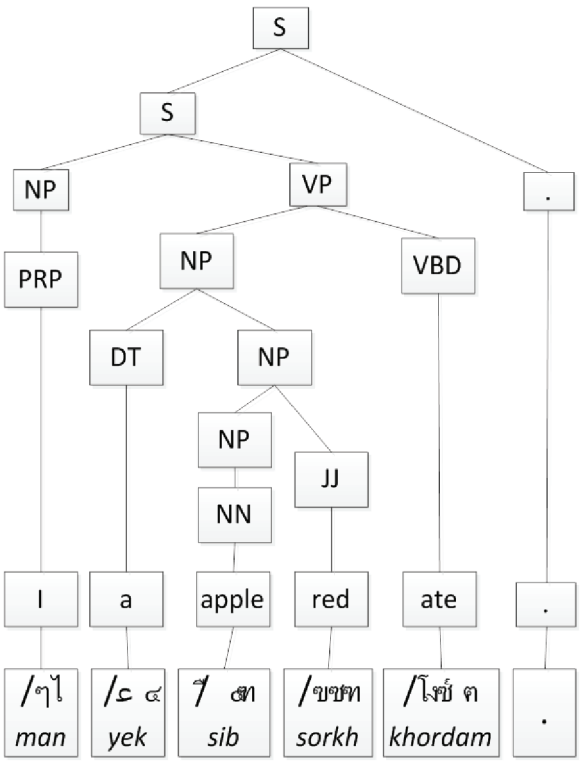

Figure 5. English derived tree with Persian word orders.

It was mentioned that our system extracts probabilistic reordering rules. After extracting all candidate reordering rules from the parallel corpus, maximum likelihood estimation method is used to calculate the probability of each reordering rule. It is obvious that for each elementary tree Ei in source language there may be more than one corresponding elementary tree with the structure of the target language and we mark them as Fij. For any rule $R: E_i(A) \rightarrow F_{ij}(A')$ which reorders and changes Ei with anchor A to Fij with the same anchor, the probability of R is calculated using Equation 1.

$$P(F_{ij} \mid E_i, A_{ik}) = \frac{Count(A_{ijk})}{Count(A_{ik})} \qquad (1)$$

Where Count (Aijk) denotes the number of Fij with A as anchor and Count (Aik) shows the frequency of Ei anchored by A in the parallel corpus.

The above process, extracts lexicalized rules. It is obvious that all elementary trees with all their possible anchors do not occur in the parallel corpus especially if the size of the corpus is not large enough. To be able to handle unseen anchors, the system uses backoff approach. It proposes some probabilistic unlexicalized rules which change E_i to F_{ij} regardless of its anchor. Like the former set of rules, the probability of this type of reordering rules is calculated as follows:

$$P(F_{ij} \mid E_i) = \frac{Count(F_{ij})}{Count(E_i)} \qquad (2)$$

In this case the anchor is ignored in calculating the frequencies and the probabilities. Some samples of the extracted rules for the elementary tree (NP (DT) (NP*))and their probabilities are listed in Table II. Table II shows that two different elementary trees with the target language word ordering are generated in our parallel corpus (some details on this corpus will be mentioned in next section): (NP@0 (DT@1) (NP*@2)) and (NP@0 (NP*@2) (DT@1)). For the elementary tree anchored by "the" 37,371 occurrences were found in our corpus, among which in 24583 the tree remains unchanged while in the remaining 12788 cases the leftmost NP is reordered with DT. Thus the probabilities of these rules are 0.84 and 0.16 respectively. These two rules are examples of lexicalized rules.

We can see in Table II that regardless of the anchors, the elementary tree (NP (DT) (NP*)) has occurred 86406 time in the corpus. In 13,929 of the cases the tree is reordered to (NP@0 (NP*@2) (DT@1)) to match the target languages word ordering while in the remaining 72,477 cases the tree stays unchanged.

TABLE II. EXTRACTED TARGET TREES FOR THE SAMPLE TREE "(NP (DT) (NP*))" WITH THEIR FREQUENCIES AND ANCHORS

F_{ij}	Anchor	Count (E_i)	Count (F_{ij})	Count (A_{ik})	Count (A_{ijk})	$P(F_{ij} \mid E_i, A_{ik})$	$P(F_{ij} \mid E_i)$
(NP@0 (DT@1) (NP*@2)) Original: originating **any action** Reordered: originating **any action**	**any**	**86,406**	**72,477**	**1,823**	**1,823**	**0.838796**	**1**
(NP@0 (DT@1) (NP*@2)) Original: That was **half a cup**. Reordered: That **half a cup** was	**half**	**86,406**	**72,477**	**48**	**36**	**0.838796**	**0.75**
(NP@0 (NP*@2) (DT@1)) Original: By **half past two** he … Reordered: By **past two half** he …	**half**	**86,406**	**13,929**	**48**	**12**	**0.161204**	**0.25**

Not all of the rules extracted with the presented method are proper to be used in rule-based systems. The collection of extracted rules may contain some noisy rules that should be eliminated before further use. The refinement process will be described in next section.

C. *Rule Refinement and Generalization*

In previous section we described how reordering rules are extracted, but not all the extracted rules are suitable to be used. After extracting the candidate rules, the refinement module eliminates some noisy rules to achieve a refined set. The following points are taken into consideration for rule refinement:

- **Elimination of non-frequent rules:** Some of the extracted rules may have high probabilities but they are extracted from few evidences. Such rules are not reliable enough and should be eliminated from the set. For this reason we have chosen a threshold for the occurrence of the elementary trees involved in each rule. If the frequency of the selected elementary tree throughout the corpus is below the threshold, the related rule is considered as "unreliable" and is eliminated. To determine this threshold a sample of the elementary trees and their extracted rules, was examined manually. The results have shown that different thresholds are suitable for different sets of trees regarding their frequencies. Table III shows these thresholds.

TABLE III. THE RELATED THRESHOLDS FOR ELIMINATION OF NON-FREQUENT RULES

Frequency	Threshold Probability
1000	.95
5000	.9
10000	.85
20000	.8
30000	.75
100000	.7
10000000	.5

- **Cumulative Elimination:** In cumulative elimination, the extracted rules are grouped regarding their elementary tree and anchor, i.e. all rules in a group are based on the same elementary tree and have the same anchor. The anchor of the group may have different translations in the target language; we show these anchors by Pz. The rules in each group are sorted according to the frequency of Pzs. The system starts from the top of the list and select the rules until it reaches a certain threshold e.g. 90% of the rules. The rest of the rules are eliminated. This threshold is set where the frequency diagram has an elbow, i.e. where the slope change, is the highest. The main goal of cumulative elimination is to compensate the alignment errors. When a specific Pz has a very low frequency, it is highly possible that the corresponding alignment which led to the rule is faulty.

- **Probabilistic Filtering:** After extracting the rules and applying the above constraints, the final probabilities are recalculated. In this stage if the probability of any rule is under a certain threshold, the rule is excluded from the final set.

It should be mentioned after each refinement step the related frequencies and probabilities are recalculated, so the next filters deal with updated information. Table IV shows some examples of the eliminated rules by the above filters.

After refining the extracted rules, our system uses a heuristic method to generalize the rules if possible. In each language we have a set of closed word classes that usually contain a fixed set of words. The rules involving anchors in these closed classes could be generalized for all the anchors in that class. Table V shows some of the closed classes used by our system.

This generalization can be applied on some open classes as well. In this system, some open classes are defined for applying generalization. The extraction of these classes and their members is a challenging task itself. In this work we selected a set of manually constructed classes to avoid further errors, thus they have low recalls. These classes include cities, animals, countries, ordinal and cardinal numbers, colors, weekdays,… For example if some rules are extracted for a satisfying number of instances in the "country" class, it could be well generalized to other countries in this class for which we haven't observed any occurrence in our train data.

TABLE IV. SOME EXAMPLES OF ELIMINATED RULES IN THE PROCCESS OF RULE REFINEMENT

Elementarytree in English	Reordered elementary tree (Persian word orders)
(VP (VP (VBD) (NP+)) (S+))	(VP@0 (S+@4) (VP@1 (NP+@3) (VBD@2)))
(VP (VP (VBD) (NP+)) (S+))	(VP@0 (VP@1 (NP+@3) (VBD@2)) (S+@4))
(VP (VP (VBG) (NP+) (SBAR+)) (VP*))	(VP@0 (VP@1 (VBG@2) (SBAR+@4) (NP+@3)) (VP*@5))
(VP (VP*) (S (VP (VB) (NP+))))	(VP@0 (VP*@1) (S@2 (VP@3 (NP+@5) (VB@4))))
(VP (VP (VBD) (S+)) (VP*))	(VP@0 (VP*@4) (VP@1 (S+@3) (VBD@2)))

V. EXPERIMENTS

In our experiments we used English and Persian as our source and target languages respectively. We used a combination of two English-Persian parallel corpora: the one presented in [13] which is constructed automatically and contains texts from some translated novels and the one presented in [14] which is gathered manually and includes both digital texts and web documents. This corpus entirely contains 11,276,318 words (89459 unique words) in the form of 770,859 sentences.

To prepare the parallel corpus for further use, some pre-processing steps were performed. These steps include unifying the encodings of the input texts. In Persian several encodings are used in electronic documents. This heterogeneity causes some problems in automatic text processing which is solved by unifying the encoding of all the texts in the parallel corpus. Moreover in the process of unifying the formats of the texts some run-on and split errors are generated, which are detected and corrected in pre-processing phase. Additionally, in many cases ZWNJs (zero width non-joiner) which are used to separate some affixes from the root words are replaced by spaces and thus a single word is modified to two separate words. These spaces are automatically replaced by ZWNJs during pre-processing.

After preparing the input texts, the sentences are parsed and the derivation and derived trees are generated. The numbers of the elementary trees are stated in Table VI. This table shows that after constructing TAG trees from parse trees of the sentences in the English side of the corpus, 11,946 different elementary trees are generated and about 6,460 trees are validated.

TABLE V. CLOSED WORD CLASSES

	ID	Class words
1	_C908	a, an
2	The	The
3	_C910	Another, other
4	_C911	Many, much, some
5	_C900	Everything, everyone, everybody
6	_C9015	This, that, those, these
7	_C9016	Each, every
8	_C933	Hmm, no, oh, wow, yes

Table VI shows that for each valid elementary tree we have on average 1.78 trees with the ordering of target language. Fig. 6 depicts the distribution of these trees (F_{ij}s).

It was clarified in previous section that rules extracted from trees with limited frequencies are considered as unreliable. The threshold used in our experiments for reliability of rules is 100, i.e. each valid rule is extracted from an elementary tree with at least 100 occurrences in the whole corpus. Among all the unique trees in the source side corpus 614 are above this threshold. The detailed statistics on the number of trees with different frequencies are mentioned in Table VII.It can be inferred from Table VII that 99% of the corpus is built by these 614 trees and the other 2,997 elementary trees only cover the remaining 1%.

The rule extraction algorithm extracted 5,794 different pairs of elementary tree and anchors, among which 5,268 English elementary trees had only one target Persian alike elementary trees, and only 526 of them had two possible Persian alike elementary trees. In these cases the rule based reordering module picks the most probable target tree for reordering. Therefore the rule extraction algorithm has extracted 5,794 lexicalized rules from the parallel corpus.

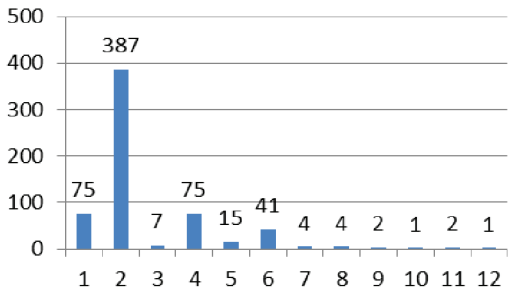

Figure 6. Distribution of F_{ij}s

TABLE VI. STATISTICS ON PARALLEL CORPUS

	Tree Count	Unique Tree Count
English elementary trees	9,945,955	11,946
English-Persian pairs	9,945,955	20,911
Valid English trees	4,557,679	6,460
Valid pairs	4,557,679	3,611

TABLE VII. CLASSIFICATION OF VALID TREES REGARDING THEIR FREQUENCIES

Tree Frequency	Count	Unique Count
1-100	38,700	2,997
101-1000	136,817	405
10001-10000	481,922	158
10000 and above	3,900,235	51
Total	**4557679**	**3,611**

It was mentioned that only 614 elementary trees remained after filtering. These trees form the 614 unlexicalized rules extracted by our system. Among the lexicalized rules, 709 rules had conflict with the unlexicalized ones. This means in these cases the anchor leads us to the correct reordering which is different from the reordering suggested by the default rule. Among these rules we can mention (NP (DT) (NP*)) anchored by "another". In this example in general (NP (DT) (NP*)) remains unchanged during reordering, but when the DT is anchored by "another" the elementary tree will be reordered to (NP (NP*) (DT)).

To evaluate the performance of the extracted reordering rules, we compare the number of crossing alignments before and after applying reordering rules to our corpus. The original sentences and the reordered ones are separately aligned with their corresponding target sentence in the parallel corpus and the number of crossing alignments is counted in each case. Smaller number of crossing alignments shows higher compatibility of reordered sentence with the orderings of target language. The results on the number of crossing alignments are stated in Table VIII.

An ideal reordering algorithm would result sentences with no crossing alignments with the target sentence. As shown in Table VIII the number of crossing alignments is decreased to 63% of its original value. Also is about 86% of the sentences the number of coring alignments is either decreased or remained unchanged but, in 13% of the sentences this number has increased. Our investigations showed that the main reasons for this increase are: Alignment errors especially for verbs and the "free-word-order" characteristic of Persian language which allows reordering of constituents in the sentence without breaking grammatical rules.

To evaluate the machine translation systems based on our reordering method, three test sets were used (Table IX shows the detailed statistics of the test sets):

- Parallel Corpus Test Set (PCTS): This set includes 400 sentences which were selected randomly from our parallel corpus and were excluded from training set. These sentences are extracted from some novels and are translated by four human translators.

TABLE VIII. STATISTICS ON THE NUMBER OF CROSSING ALIGNMENTS BEFORE AND AFTER REORDERING

Number of sentence	**770,860**
Number of crossing alignments before applying reordering rules	17,551,756
Number of crossing alignments after applying reordering rules	11,205,633
Number of sentences with decreased number of crosses	428,358
Number of sentences with increased number of crosses	107,849
Number of sentences with no changes in the number of crosses	234,652

TABLE IX. STATISTICS ON TEST SET

	PCTS	**EGIU**	**News**
Sentence Count	400	2522	820
Word Count	5501	16384	22192
Unique Word Count	1582	2242	4735

- English Grammar In Use (EGIU): This set is selected from "English Grammar In Use" which is a book for self-studying English language. This book covers almost all the common structures in English sentences. This set was translated by two human translators for evaluation purpose.
- News: This set contains 819 sentences (22,187 words) from different news websites. These sentences were translated by four human translators [15].

To evaluate our reordering method, we constructed two different machine translators which use our reordering method and compared their results with each other and also with three other translators. These translators are listed below:

- A conventional SMT system
- A hybrid SMT system in which the source sentences are reordered using our method and then the reordered sentences are translated with a monotone SMT model.
- A hybrid SMT system in which the source sentences are reordered using our method and then the reordered sentences are translated with a conventional SMT model.
- A rule-based machine translator introduced in [21] which also uses TAG as its formalism.
- Google translator.

Although the training set of our translators are different from Google translator but, as this translator is free and publically available, we included it in our comparisons. Fig. 7 shows the test results on EGIU test set. In this case our reordering method combined with monotone SMT achieved the best results. In these tests all the translators except Google's were close in accuracy and our model improved the BLEU score by 0.4 points regarding the conventional SMT system which is our baseline.

Fig. 8 depicts the results on PCTS set. In this series of tests our reordering method combined with conventional SMT brings the best results. As can be seen in this figure, the BLEU score in best case is around 1.8 points more than the conventional SMT model. In combination with monotone SMT, our results are 0.55 points above the baseline.

Test results on News dataset is illustrated in Fig. 9. In these tests Google translator outperforms the others. It seems Google's training set mostly includes news articles and that is why it performs pretty well in translating news articles. Apart from Google, conventional SMT enhanced by our reordering method, improved the BLEU score by 1.45 points regarding the baseline.

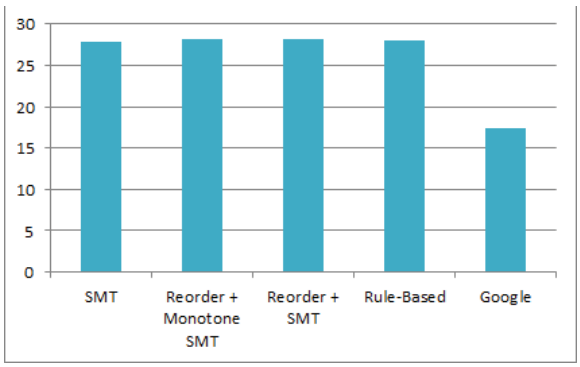

Figure 7. Test results on EGIU corpus using BLEU score

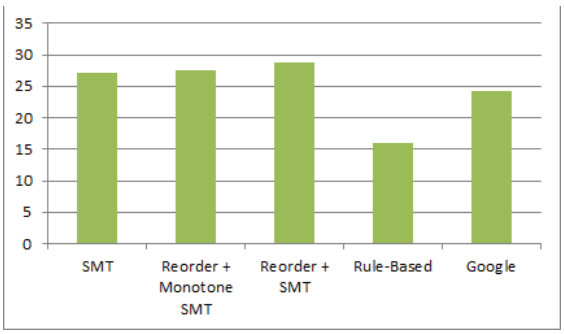

Figure 8. Test results on PCTS corpus using BLEU score

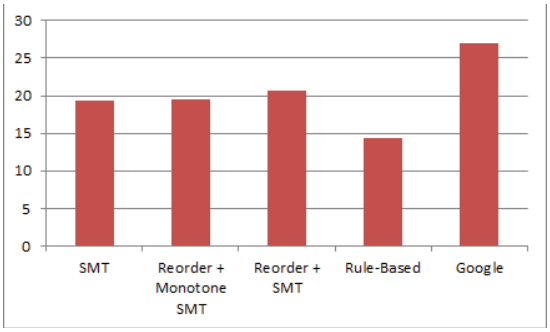

Figure 9. Test results on News corpus using BLEU score

It could be inferred from Fig. 8 and Fig. 9 that the combination of the reorderings proposed by our system and the ones suggested by the conventional SMT system, has improved the quality of translation. It means our system performs some reordering which conventional SMT is unable to detect and vice versa. Closer looks on test results showed that conventional SMT, using its phrase table, can perform local reordering better than our rule-based reordering method while our method is strong in detecting long-distance or global reorderings.

VI. Conclusion and Future Work

In this paper a hybrid machine translation algorithm was presented. This approach tries to alleviate the shortcomings of statistical method in detecting long distance dependencies and linguistics constituents. For this reason synchronous TAG was selected as the grammar formalism because of its abilities in modeling natural language properties.

In first step the system automatically extracts a set of reordering rules to be used in the rule based module. These rules are exploited to reorder the words of source sentence to be similar to target language orderings. The extracted rules can be well utilized in pure rule based translators. In translation phase the given sentence passes through the rule based module and its structure is changed to match the properties of the target language. This sentence is then fed to statistical translator to be translated by conventional statistical approach.

In future work elementary trees could be used as a factor in factor-based translation models. Furthermore the rule extraction algorithm could be enhanced by using a dictionary. The scope of extracted rules could be changed to phrases instead of words which will be more suitable for using in phrase based SMT models. Also as conventional phrase-based SMT is proved to be strong in local reorderings, our method could be changed in order to only detect long-distance reorderings and leave the local ones to phrase table.

References

[1] F. J. Och and H. Ney, “Statistical machine translation,” In Proceedings of the European Association for Machine Translation Workshop (EAMT), 2000, pp. 39-46.

[2] T. P. Nguyen, A. Shimazu1, T. Ho, M. Nguyen1, and V. V. Nguyen, “A Tree-to-String Phrase-based Model for Statistical Machine Translation”, In Proceedings of the 12th Conference on Computational Natural Language Learning, Manchester, 2008, pp. 143–150.

[3] K. Yamada and K. Knight, “A Syntax-based Statistical Translation Model”, InProceedings of the 39th Annual Meeting on Association for Computational Linguistics (ACL '01). Association for Computational Linguistics, Stroudsburg, PA, USA, 2001, pp. 523-530.

[4] M. Galley, M. Hopkins, K. Knight, and D. Marcu, “What’s in a Translation Rule”, In Proceedings of The Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL- HLT), Boston, Massachusetts, USA, 2004, pp. 273—280.

[5] F. Xia and M. McCord, “Improving a Statistical MT System with Automatically Learned Rewrite Patterns”, In Proceedings of the 20th International Conference on Computational Linguistics (COLING 2004), Association for Computational Linguistics, Stroudsburg, PA, USA, 2004, Article 508.

[6] M. Popović and H. Ney, “POS-based Word Reorderings for Statistical Machine Translation”. In Procedeengs of International Conference on Language Resources and Evaluation (LREC), pages 1278-1283, Genoa, Italy, 2006, pp. 1278–1283.

[7] N. Ueffing and H. Ney, “Using POS Information for Statistical Machine Translation into Morphologically Rich Languages”. In Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics (EACL), Budapest, Hungary, 2003, pp. 347-354.

[8] S. DeNeefe, “TREE-ADJOINING MACHINE TRANSLATION”. PhD thesis, University Of Southern California, December 2011.

[9] Y. Liu, Q. Liu and Y. L¨u, “Adjoining Tree-to-String Translation”, In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics, Portland, Oregon, 2011, pp. 1278–1287.

[10] S. M. Shieber and Y. Schabes. “Synchronous tree-adjoining grammars”. In Proceedings of the 13th International Conference on Computational Linguistics (COLING 1990), pp. 253-258.

[11] A. K. Joshi, L. S. Levy, and M. Takahashi. “Tree adjunct grammars”. Journal of Computer and System Sciences, Vol. 10, Issue 1, 1975, pp. 136–163.

[12] F. Xia, “Automatic grammar generation from two different perspectives,” PhD thesis, University of pennsylvania, 2001.

[13] Mansouri, A. and H. Faili, “State-of-the-art English to Persian Statistical Machine Translation System”. In Proceedings of the 16th CSI International Symposiums on Artificial Intelligence and Signal Processing (AISP 2012), 2012.

[14] Miangah, T. M. “Constructing a large-scale English-Persian parallel corpus” Meta: Translators’ Journal, vol. 54, Issue 1, 2009, pp. 181–188.

[15] F. Jabbari, S. Bakhshaei, S. M. Mohammadzadeh Ziabary, S. Khadivi, "Developing an Open-domain English-Farsi Translation System Using AFEC: Amirkabir Bilingual Farsi-English Corpus", Association for Machine Translation in the Americas (AMTA 2012) , 2012.

[16] Zens, R., F. J. Och, H. Ney, “Phrase-based statistical machine translation,” KI 2002: Advances in Artificial Intelligence, 2002, pp. 35-56.

[17] K. Rottmann and S. Vogel, "Word reordering in statistical machine translation with a pos-based distortion model," In Proceedings of the 11th International Conference on Theoretical and Methodological Issues in Machine Translation (TMI-11), Skvde, Sweden, 2007, pp. 171-180.

[18] D. Klein and Ch D. Manning, "Accurate Unlexicalized Parsing," In Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1 (ACL '03), Vol. 1. Association for Computational Linguistics, Stroudsburg, PA, USA, 2003, pp. 423-430.

[19] F. J. Och, H. Ney, "Improved Statistical Alignment Models". In Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics, Hongkong, China, 2000, pp. 440-447.

[20] S. M. Shieber. Probabilistic synchronous tree-adjoining grammars for machine translation: The argument from bilingual dictionaries. In Proceedings of the Workshop on Syntax and Structure in Statistical Translation (SSST, NAACL-HLT), Rochester, New York, 2007,pp. 88–95 .

[21] H. Faili and G. Ghassem-Sani. "An Application of Lexicalized Grammars in English-Persian Translation",In Proceedings of the 16th European Conference on Artificial Intelligence (ECAI 2004), Valencia, Spain, 2004, pp. 596-600.

[22] M. Carpuat, Y. Marton and N. Habash, "Improved Arabic-to-English statistical machine translation by reordering post-verbal subjects for word alignment", Machine Translation, March 2012, Volume 26, Issue 1-2, pp. 105-120.

[23] N. Bach,"Dependency Structures for Statistical Machine Translation". PhD thesis, Carnegie Mellon University, 2012.

[24] D. N. Mehay, and C. Brew,"CCG Syntactic Reordering Models for Phrase-based Machine Translation". Proceedings of WMT-2012, Montreal, Canada, 2012.

[25] H Almaghout, "CCG-Augmented Hierarchical Phrase-Based Statistical Machine Translation", Phd thesis, Dublin City University, July 2012.

[26] M. Collins and P. Koehn, "Clause restructuring for statistical machine translation",In Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics, 2005, pp. 531–540.

[27] S. Howlett and M. Dras, "Dual-path phrasebased statistical machine translation", In Proceedings of the Australasian Language Technology Association Workshop, 2010, pp. 32–40.

[28] N. Habash,"Syntactic preprocessing for statistical machine translation", In Proceedings of the MT Summit XI, 2007, pp. 215–222.

[29] C. Wang, M. Collins, and P. Koehn. "Chinese syntactic reordering for statistical machine translation". In Proceedings of the 2007 Joint Conference on EmpiricalMethods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), 2007, pp. 737–745.

[30] Y. Liu, Q. Liu, and S. Lin, "Tree-to-String Alignment Template for Statistical Machine Translation", In Proceedings of the 21st International Conference on Computational Linguistics (ACL 06), Sydney, July 2006, PP. 609–616.

Amin Mansouri received his B.Sc. degree in software engineering from Razi University and his M.Sc. degree in software engineering from University of Tehran at 2012. His research interests include computational natural language processing and machine translation.

Hakimeh Fadaei received her B.Sc. and M.Sc. degrees in computer engineering from Shahid Beheshti University. She is currently a Ph.D. student on software engineering at University of Tehran. Her research interests include natural language processing, text mining and information retrieval.

Heshaam Faili received his B.Sc. and M.Sc. degrees on software engineering and Ph.D. degree in artificial intelligence from Sharif University of Technology. He is an assistant professor at University of Tehran in the school of Electrical and Computer Engineering. His research interests include natural language processing, machine translation, data mining and social networks.

Mohsen Arabsorkhi received his B.Sc. degree in software engineering from Shahid Beheshti University and his M.Sc. degree from Shiraz University. He is a lecturer at Islamic Azad University and works as a researcher at the NLP lab in University of Tehran. His research interests include natural language processing, machine translation and first language acquisition.