A Three-Layer Architecture for Automatic Post-Editing System Using Rule-Based Paradigm

Mahsa Mohaghegh
Unitec
Auckland, New Zealand
mmohaghegh@unitec.ac.nz

Abdolhossein Sarrafzadeh
Unitec
Auckland, New Zealand
hsarrafzadeh@unitec.ac.nz

Mehdi Mohammadi
SheikhBahaee University
Isfahan, Iran
mehdi.mka@gmail.com

Abstract

This paper proposes a post-editing model in which our three-level rule-based automatic post-editing engine called Grafix is presented to refine the output of machine translation systems. The type of corrections on sentences varies from lexical transformation to complex syntactical rearrangement. The experimental results both in manual and automatic evaluations show that the proposed system is able to improve the quality of our state-of-the-art English-Persian SMT system.

1 Introduction

The overall success of well-designed statistical machine translation (SMT) systems has made SMT one of the most popular machine translation (MT) approaches (Callison-Burch, Koehn, Monz, & Zaidan, 2011). Currently however, MT output is often seriously grammatically incorrect. This is more often the case in SMT than other approaches due to the absence of linguistic rules for the language pair on which it is being applied. Grammatical error not only weakens the fluency of the translation, but in certain cases it completely changes the meaning of a sentence. In morphologically rich languages, grammatical accuracy is of more significance, as the interpretation of syntactic relations depends heavily on morphological agreements within the sentences. Since our system's approach is SMT, and deals with Persian, a morphologically rich language, post-editing the output is an important step in maintaining the fluency of the translation, and as we will show, this can yield to higher evaluation scores and more fluent translation. Due to the repetitive nature of machine translation mistakes (Allen & Hogan, 2000) and the similarity of automatic post-editing (APE) process to machine translation process (Simard, Goutte, & Isabelle, 2007) certain MT systems can carry out the task of an APE component.

The SMT approach to MT is useful in that it operates on numerical data extracted from parallel corpus. This approach tends to reduce human cost at translation stage. However, the search cost is expensive, and the system has no linguistic background (although generally it is this that enables the system to be applied to any language pair after training on language-specific data). Most systems of this approach also encounter difficulties when capturing long distance phenomena.

RBMT approaches are categorized under three different types: Direct Systems, Transfer RBMT Systems that employ morphological and syntactical analysis, and Interligual RBMT Systems that use an abstract meaning. The Grafix APE system follows the pattern of Transfer-based systems. The aim of development of Grafix system is to correct some grammatical SMT system errors frequently occur in English-to-Persian translations.

2 Related Works

According to Xuan, Li, and Tang (2011) the most popular combinations of MT systems and APE modules are a RBMT main system linked with phrase-based SMT (PB-SMT) and a RBMT main system linked with Example-based MT (EBMT). Simard et al. (2007) and Lagarda, Alabau, Casacuberta, Silva, and Diaz-de-Liano (2009) and Isabelle, Goutte, and Simard (2007) have used a phrase-based SMT to enhance the output of an RBMT system. Pilevar (2011), in his recent work used a statistical post-editing (SPE) approach to improve the translation of subtitles for movies for the English-Persian language pair. The author notes that in contrast to same reported experiments in other languages, their result for the combination of RBMT+SPE is slightly weaker than SMT alone.

The 4th Workshop on South and Southeast Asian NLP (WSSANLP), International Joint Conference on Natural Language Processing, pages 17–24,
Nagoya, Japan, 14-18 October 2013.

Ahsan, Kolachina, Kolachina, Sharma, and Sangal (2010) document work using a modular RBMT system which is able to combine the two translation approaches at different stages in the RBMT system's pipeline. In this way, exploration of rules for both local and long distance reordering could be performed independently, and such reordering leading to improved translation output could be identified and utilized. In their paper, the authors show an increase in the output score for each stage of combination.

Recently however, there has been more interest in the use of RBMT as the base for an APE system. Marecek, Rosa, and Bojar (2011) report their experiments in correcting the output of an English-Czech MT system. They performed a rule-based grammatical correction module, DEPFIX, on sentences parsed to dependency trees. Their baseline SMT system relies on Moses, a phrase-based translation system. Their two-step translation is a setup in which the English source is translated into simplified Czech. Then the simplified Czech is monotonically transferred to a fully inflected Czech. In 2012, Rosa, Marecek, and Duˇsek (2012) enhanced DEPFIX by enriching the rule set and using a modified version of MST Parser. These two modifications, based on their results, led to higher scores.

3 Description of the System

In this study, we couple the PeEn-SMT system previously developed by Mohaghegh, Sarrafzadeh, and Moir (2011) with an RBMT-based APE. The architecture of our system differs from common approaches like Simard, et al. (2007) and Lagarda, et, al. (2009). Because of its use of language and translation models over large corpora, together with better lexical selection, SMT would be capable of consistently providing fluent translation. This approach, however, lacks linguistic knowledge, which we believe is more important when it comes to an APE. In this area we believe RBMT is a superior base due to its strengths in linguistic knowledge.

Three levels of transformations as shown in Figure 1, constitute the overal design of the Transfer-based APE module: lexical transformers, shallow transformers and deep transformers. The output of the SMT system is passed to the APE as input. This is then run through a series of transformers and fine-tuned in an effort to achieve a more accurate translation.

3.1 The Underlying SMT System

Persian is a morphologically rich language, so word disordering is a common issue that we face. Hierarchical SMT (D. Chiang, 2005) as an extension to phrase-based translation takes syntax into account to some extent, with phrases being used to learn word reordering. This improvement is due to the word order differences between Persian and English, which are better handled with a hierarchical phrase based system than a standard phrase-based approach. These advantages of hierarchical SMT encouraged us to conduct our study on Joshua after numerous experiments with Moses.

Hierarchical phrase-based translation allows phrases with gaps to be modeled as synchronous context-free grammars (SCFGs). In effect, it is grammars that are used, not phrase tables. The hierarchical approach does not detract from the strengths of phrase-based approaches, but uses them to its advantage. Phrases are used in order to learn word reordering. In a hierarchical approach, this principle is taken a step further, and phrases are used for *phrase* reordering, using SCFGs to compose the hierarchical phrases from words and sub-phrases and represent translation models.

Joshua (Li, Callison-Burch, Khudanpur, & Thornton, 2009), a hierarchical phrase based machine translation toolkit, was a reimplementation of the Hiero MT system (D. Chiang, 2007), and is able to support formalisms such as SAMT

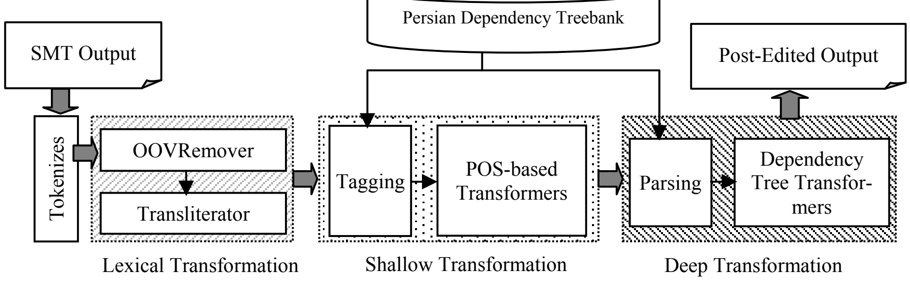

Figure 1. Overall architecture of the proposed Rule-based APE system

(Zollmann & Venugopal, 2006). More recent efforts introduced the Thrax module, an extensible Hadoop-based extraction toolkit for synchronous context free grammars (Weese, Ganitkevitch, Callison-Burch, Post, & Lopez, 2011).

In Joshua, an SCFG can be represented as a set of rules given as:

$$C_i \rightarrow \langle \alpha_i , \gamma_i , \sim_i , \varphi_i \rangle \quad (1)$$

where C_i is a non-terminal symbol of the grammar, α_i and γ_i are sequences of terminal and non-terminal symbols for the source and target sides respectively, $\sim_i$ is a correspondence between the non-terminals of α_i and γ_i, and φ_i is a feature vector defining the probability of translation from α_i to γ_i.

3.2 Training Data Source

Grammatical relations may be represented by dependency structures. Compared to syntactic trees, dependency structures are more specific in regard to semantics rather than strict word order (Ambati, 2008). The structure of sentence dependency tree represents the relationship between words as either modifying or being modified, and the root in each tree being the only word which does not modify any other word.

The Persian Dependency Treebank[1], containing over 12500 sentences and 189000 tokens, is our main source of training data for POS-tagging and dependency parsing. Its data format is based on CoNLL Shared Task on Dependency Parsing (Buchholz & Marsi, 2006).

3.3 Pre-Processing and Tagging

We pre-process input Persian sentences using our implemented tokenizer component. It eliminates whitespace, semi-space, tab and new line in order to tokenize the text. We also considered punctuation marks in sentences in cases where punctuation marks were attached to other words.

POS-tagging the input text is a pre-requisite process for both shallow and deep transformation levels. The Maximum Likelihood Estimation (MLE) is the underlying algorithm in the POS-tagger component for our APE method and is trained with the Persian Dependency Treebank data. Evaluation showed that the use of this approach for tagging the Persian language yielded promising results (Raja et al., 2007). Tagging tests showed the best accuracy to be 95.43%.

3.4 Parsing

There has recently been increasing interest in the use of dependency models for a number of applications in NLP, particularly since certain characteristics of the dependency structure prove to be advantageous over other syntactic representations (Ding & Palmer, 2005). In a dependency parsing tree, words are linked to their arguments by dependency representations (Hudson, 1984).

We used an implementation of Dependency Parsing named MSTParser. MSTParser's main algorithm is Maximum Spanning Tree in which the maximum spanning tree should be found in order to find the best parse tree (Kübler, McDonald, & Nivre, 2009).

3.5 Three-Level Rule-based Transformers

We acquired translation rules manually by investigating a broad range of incorrect translations and determining frequent wrong patterns among them. In order to determine incorrect patterns and define correction rules for them, it is necessary to parse both the MT output and the reference text.

By investigating the incorrect and incomplete translation outputs and considering the dependency parser output for these sentences, a number of incorrect patterns were identified in the POS sequence and Dependency parse tree of these sentences. These patterns are compared against the Persian Dependency Treebank to ensure that they do not match a known correct sequence.

Lexical Transformation: Two components are serving in the first level of transformations. OOV[2] remover is a substitution rule like $E \rightarrow F_1, F_2 \ldots F_n$ where E is an English word and $F_i = F_1, F_2 \ldots F_n$ are different translations of the word in Persian. This rule replaces a remained English word in the MT output with the correct translation in Persian. Since no Word Sense Disambiguation (WSD) component is present, it is assumed that the first meaning found for the English word in the dictionary used is the most frequent translation of the word, so it is used as replacement for the English word. A transliterator also is used to complement the operation of OOV remover in cases that

[1] http://dadegan.ir/en

[2] Out Of Vocabulary

OOV remover could not find any equivalent Persian translation for English words in the output. A transliterator works based on training an amount of prepared data to produce the most likely Persian word for the English word remaining in the sentence. The result is the English word appearing composed with Persian character scripts. The training data set for transliterator contains over 4600 of the most frequently used Persian words and named entities written using English letters, and also the equivalent in Persian script. In order to implement the transliterator component we used some libraries from Virastyar[3] software.

Shallow Transformers: The second stage of the system involves a shallow transfer module which is based on some POS patterns identified as incorrect. Incorrect or incomplete POS sequence patterns will be controlled for each sentence, and appropriate rules executed to revise them. Description of some shallow rules comes in following section.

IncompleteDependentTransformer: Relative pronouns such as «که» in Persian (English *"that"*) which POS-tagged as SUBR, suggest continuation of the phrase by a dependent clause. If it occurred in a POS sequence without a consequent verb, an incomplete dependent sentence would be identified and a verb should complete the sentence. Currently, in most instances the verb «است» (English *"is"*) is suggested.

IncompleteEndedPREMTransformer: Pre-modifiers (denoted by PREM) are noun modifiers. According to the definition, modifiers should precede nouns, so a POS sequence in which a pre-modifier located at the end of a sentence deemed as incorrect. These sequences were removed from the sentence altogether since there is no logical translation for given input.

AdjectiveArrangementTransformer: In the Persian language, adjectives usually come after the nouns they describe. For instance (English *"heavy bag"*) «کیف سنگین» is translated literally as *"bag heavy"*. The only exception in this group is superlative adjectives, which are identified by the suffix «ترین» attached to the adjective. In this special case, the adjective comes before the noun to define it. The appearance of non-superlative adjectives before their described nouns indicates incorrect composition which must be corrected by this transformer.

Deep Transformers: In this type of transformations, the input is parsed by a dependency parser. Once the text is POS-tagged, some preparation is performed to parse the input, according to the parsing input format (McDonald, Pereira, Ribarov, & Hajic, 2005). The rules here examine dependency tree of each sentence to verify it bounds to some syntactical and grammatical constraints. The tree structure of each parsed sentence will be analyzed, and those sentences with incorrect parse tree structures will be evaluated to determine if correction is necessary. Some deep rules are described in sections below.

NoSubjectSentenceTransformer: Compared to known translation reference sentences, it was seen in some cases that what was parsed as the object in the sentence was actually the subject. Such sentences have a third person verb, no definite subject and an object tagged as POSTP (postposition) in the POS sequence. This transformer is designed to revise the sentence by removing the postposition «را» which is the indicator of a direct object in the sentence. Removal of this postposition changes the sentence to one with a subject.

PluralNounsTransformer: Unlike English, in the Persian language the word coming after a number is always singular. In SMT output there are instances where plural nouns are located after a number (< PRENUM> POS). This is corrected by removing the plural symbol of Persian words. The suffix«ها» (/hã/) is the most common plural indicator, which is removed in this rule.

VerbArrangementTransformer: Persian language has a preferred word order, with SOV (subject-object-verb) followed by SVO. These two types make up more than 75% of natural languages having a preferred order (Crystal, 2004) . Although reordering of sentence components does not necessarily lead to a significant change in meaning, there are many cases where these changes may disturb the fluency and accuracy of the sentence. One frequently occurring case is sentences in which a main verb as Root does not occur immediately before the period punctuation. The transformer treats such cases as follows: the sentence is

[3] http://sourceforge.net/projects/virastyar/

reordered by moving the root verb and its NVE[4] dependants (in the case of compound verbs) to the end of the sentence, just before the period punctuation mark.

MissingVerbTransformer: Sometimes sentences from SMT output can occur with missing verbs specifically in the case of compound verbs. We used the Persian [verb] Valency Lexicon (Rasooli, Moloodi, Kouhestani, & Minaei-Bidgoli, 2011) to determine the proper verb for a non-verbal element in the sentence. All obligatory and optional non-verbal elements (main-verb dependents) are listed in this lexicon. For example, the verb «مصرف کردن» (English *"to consume"*) is composed of two elements. Searching for «مصرف» in this lexicon will return «کردن» as the main part of that particular compound. We find the correct root of the verb (past or present) by examining the other verbs in a sentence which could show the correct tense intended for that sentence. The present tense is chosen by default if there are no other verbs in the sentence. Any subject with a referred verb preceding the subject in the dependency tree is identified as an incorrect linked subject to any verb due to violence of standard SOV structure. In this case, the last word in the sentence is considered as a candidate of non-verbal element in the Verb Valency Lexicon. If a corresponding verb is found, the correct tense of that verb will be inserted into the space of the missing verb.

4 Experiments and Results

4.1 Baseline SMT

We used Joshua 4.0 as our baseline system using default settings. Training data was extracted from a parallel corpus based on the NSPEC corpus tested by (Mohaghegh & Sarrafzadeh, 2012). After some modification to remove inconsistencies in the translation output, the final corpus (names NPEC) consisted of almost 85,000 sentence pairs of 1.4 million words, originating mostly from bilingual news sites. There are a number of different domains covered in this corpus, but the majority of the texts were in literature, science and conversation. The language model used in the tests was extracted from IRNA[5] website, covering news stories, and comprised over 66 million words.

4.2 Test Data Set

Our evaluation is based on eight test sets extracted from certain bilingual websites. Randomly selected, test sentences cover different domains such as art, medicine, economics and news. The direction of translation is English-Persian. In the evaluation process, the Persian side of the test sets was used as the translation reference to score the output quality of both the baseline system and the final post-APE output. The size of the test data varies from one paragraph of text (more than 100 words) to a complete page or more (up to 600 words). The number of sentences in both sides is equal. The total number of test sentences is 153. The reason the number of test sentences was less than what would normally be preferred is that finding perfectly correct human translations for sentences for scoring purposes covering a number of domains is a very difficult task and manual evaluation by annotators would be much more labor-intensive.

4.3 Automatic Evaluation

We used both BLEU and NIST to evaluate the effect of APE system on translation quality. The results of translation before and after APE are shown in Table 1.

As Figures 2 and 3 also demonstrate, the results generally show increases in both metrics. The greatest increase in BLEU score due to the APE was achieved in test set #3, with an increase of about 0.15 BLEU, while the greatest NIST score increase was in test set #1, with a 0.16 increase.

However, in certain test sets the scoring metrics report a decrease in output quality, the worst BLEU score being at a difference of -0.0151, and the worst NIST at -0.27. The main reason behind weakened results is lack of training data for the Transliterator module, as it scripted some proper names and terms incorrectly in Persian. The large difference between the BLEU scores of data sets is due to each data set genre and the type of training data set. The quality of statistical translation (in terms of BLEU metric score) affects the APE module directly. The test set is in the news story domain, the same domain as the parallel corpus

[4] Non-Verbal Element: Many Persian verbs are compound verbs. They consist of two parts; one verbal and one non-verbal element.

[5] http://www.irna.ir/ENIndex.htm

Input	Size(words)		Before APE		After APE		BLEU Difference	NIST Difference
	En	Fa	BLEU	NIST	BLEU	NIST		
#1	163	158	0.6523	6.5740	0.6770	6.7349	0.0247	0.1609
#2	218	222	0.2232	1.0870	0.2187	1.0935	-0.0045	0.0065
#3	371	403	0.5914	6.1083	0.7388	6.1089	0.1474	0.0006
#4	362	337	0.1365	0.7962	0.1214	0.7064	-0.0151	-0.0898
#5	101	115	0.7925	5.7332	0.8716	5.4624	0.0791	-0.2708
#6	354	386	0.2738	1.8922	0.2779	1.9196	0.0041	0.0274
#7	555	653	0.2945	2.0457	0.2951	2.0333	0.0006	-0.0124
#8	259	297	0.4048	2.3940	0.4089	2.4052	0.0041	0.0112

Table 1. Scores of APE based on SMT Joshua version 4.0

used. In test set #4, in the religious genre, the decrease in both BLEU and NIST is attributed to a lack of data in this genre.

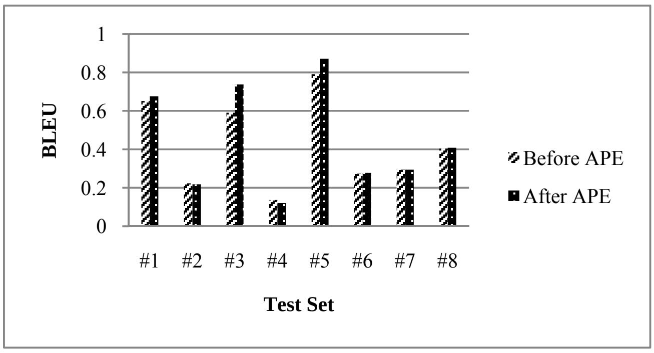

Figure 2. Difference of BLEU score after applying APE on eight test sets

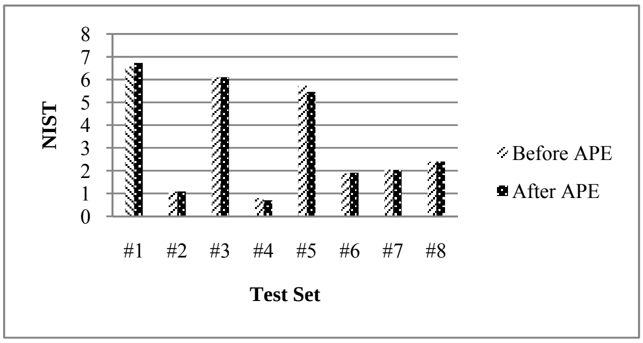

Figure 3. Difference of NIST score after applying APE

4.4 Manual Evaluation

As suggested by (Marecek et al., 2011), grammatical correctness of sentences cannot be measured appropriately with BLEU metrics. Because of this, we evaluated the output manually. The same test sets in automatic evaluation containing 153 sentences are evaluated here. We assigned the APE output of test sentences to two separate annotators, instructing them to rank the APE output sentences based on whether the output showed improvement, decrease, or no change in fluency compared to the original SMT output. Table 2 summarizes the results of this manual evaluation.

Annotator/ Rank	**Improved**	**No Change**	Weakened
Annotator 1	47	95	11
Annotator 2	43	99	11

Table 2. Manual evaluation scores for 153 test sentences

Both annotators, who completed the evaluation without discussing or consulting with each other, had very similar judgment of the APE system's output. The results show that the APE system has been successful in improving the quality of the baseline SMT system output by 29.4%. The current developed rules for the APE system are effective for about 37% of the SMT translated sentences.

We also identified both annotator agreements on ranked sentences. Based on their agreement, the quality improvement from the APE system is 25% and weakened by 3%. The comparison of the average percent of manual scores is shown in Figure 4.

5 Conclusion and Future Work

We presented an uncommon automatic post-

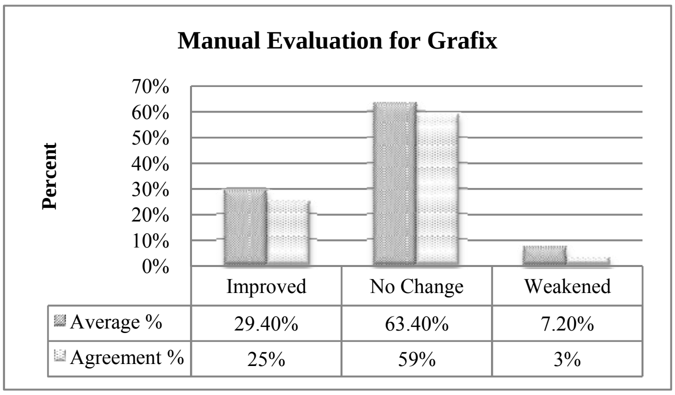

Figure 4. Manual evaluation of 153 test sentences

editing model for English-Persian statistical machine translation developed using a rule-based approach in different levels of transformation. The automatic evaluation results in terms of BLEU metric show that 75% of the test sets have been improved as far as the quality of the translation is concerned after post-editing. Although is affected by the quality of SMT system output, our APE approach is shown to be able to improve the SMT output up to 0.15 BLEU. On the other hand we faced some decrease in quality of translation by applying the system. We found that these results came from the lexical transformer level. We believe that this is due to OOV words remaining in the original script, and that the application of OOVRemover and Transliterator only produced a new unknown (incorrect) word as the original word equivalent. This phenomenon has decreased the BLEU score in one case by up to -0.015 BLEU.

Our results in terms of NIST and BLEU show that automatic evaluation could not reveal the quality of grammatical changes appropriately. However, manual evaluation scores show that a rule-based approach for an APE system is useful for improving at least 25% of the translation with a loss of at most 7%. To increase the improvement and decrease the loss of accuracy, we intend to enrich the bilingual dictionary used in OOVRemover as well as training data for Transliterator. Extending the rules in both shallow and deep levels is another task we plan to focus on. We also intend to investigate the use of an automatic rule induction module using text mining or similar approaches.

References

Ahsan, A., Kolachina, P., Kolachina, S., Sharma, D. M., & Sangal, R. (2010). Coupling statistical machine translation with rule-based transfer and generation. *In Proceedings of the 9th Conference of the Association for Machine Translation in the Americas*. Denver, Colorado.

Allen, J., & Hogan, C. (2000). Toward the Development of a Post-editing Module for Raw Machine Translation Output: A Controlled Language Perspective. *Third International Controlled Language Applications Workshop (CLAW-00)* (pp. 62-71).

Ambati, V. (2008). Dependency Structure Trees in Syntax Based Machine Translation. In Advanced MT Seminar Course Report.

Buchholz, S., & Marsi, E. (2006). CoNLL-X shared task on multilingual dependency parsing. *In Proceedings of the Tenth Conference on Computational Natural Language Learning* (pp. 149-164). New York City, USA: Association for Computational Linguistics.

Callison-Burch, C., Koehn, P., Monz, C., & Zaidan, O. F. (2011). *Findings of the 2011 workshop on statistical machine translation*.

Chiang, D. (2005). A hierarchical phrase-based model for statistical machine translation *In Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics* (pp. 263-270). University of Michigan, USA.: Association for Computational Linguistics.

Chiang, D. (2007). Hierarchical phrase-based translation. *Computational Linguistics, 33*(2), 201-228.

Crystal, D. (2004). The Cambridge Encyclopedia of English Language: Ernst Klett Sprachen.

Ding, Y., & Palmer, M. (2005). Machine translation using probabilistic synchronous dependency insertion grammars *In Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics* (pp. 541-548). Ann Arbor, USA: Association for Computational Linguistics.

Hudson, R. A. (1984). Word grammar: Blackwell Oxford.

Isabelle, P., Goutte, C., & Simard, M. (2007). Domain adaptation of MT systems through automatic post-editing *In Proceedings of Machine Translation Summit XI* (pp. 255-261). Phuket, Thailand.

Kübler, S., McDonald, R., & Nivre, J. (2009). Dependency parsing *Synthesis Lectures on Human Language Technologies* (Vol. 1, pp. 1-127).

Lagarda, A. L., Alabau, V., Casacuberta, F., Silva, R., & Diaz-de-Liano, E. (2009). Statistical post-editing of a rule-based machine translation system. *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers* (pp. 217-220). Boulder, Colorado: Association for Computational Linguistics.

Li, Z., Callison-Burch, C., Khudanpur, S., & Thornton, W. (2009). Decoding in Joshua. *Prague Bulletin of Mathematical Linguistics, 91*, 47-56.

Marecek, D., Rosa, R., & Bojar, O. (2011). Two-step translation with grammatical post-processing. *In Proceedings of the Sixth Workshop on Statistical Machine Translation* (pp. 426-432). Stroudsburg, PA,USA: Association for Computational Linguistics.

McDonald, R., Pereira, F., Ribarov, K., & Hajic, J. (2005). Non-projective dependency parsing using spanning tree algorithms. *In Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing* (pp. 523-530). Vancouver, B.C., Canada: Association for Computational Linguistics.

Mohaghegh, M., & Sarrafzadeh, A. (2012). *A hierarchical phrase-based model for English-Persian statistical machine translation*.

Mohaghegh, M., Sarrafzadeh, A., & Moir, T. (2011). *Improving Persian-English Statistical Machine Translation: Experiments in Domain Adaptation*.

Pilevar, A. H. (2011). USING STATISTICAL POST-EDITING TO IMPROVE THE OUTPUT OF RULE-BASED MACHINE TRANSLATION SYSTEM. *Training, 330*, 330,000.

Raja, F., Amiri, H., Tasharofi, S., Sarmadi, M., Hojjat, H., & Oroumchian, F. (2007). Evaluation of part of speech tagging on Persian text. *University of Wollongong in Dubai-Papers*, 8.

Rasooli, M. S., Moloodi, A., Kouhestani, M., & Minaei-Bidgoli, B. (2011). A syntactic valency lexicon for Persian verbs: The first steps towards Persian dependency treebank. *In Proceedings of 5th Language & Technology Conference (LTC): Human Language Technologies as a Challenge for Computer Science and Linguistics* (pp. 227-231).

Rosa, R., Marecek, D., & Duˇsek, O. (2012). DEPFIX: A System for Automatic Correction of Czech MT Outputs. *In Proceedings of the Seventh Workshop on Statistical Machine Translation,* . Montreal, Canada: Association for Computational Linguistics.

Simard, M., Goutte, C., & Isabelle, P. (2007). Statistical phrase-based post-editing *In Human LanguageTechnologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference* (pp. 508-515). Rochester, USA.

Weese, J., Ganitkevitch, J., Callison-Burch, C., Post, M., & Lopez, A. (2011). *Joshua 3.0: Syntax-based machine translation with the thrax grammar extractor*.

Xuan, H., Li, W., & Tang, G. (2011). An Advanced Review of Hybrid Machine Translation (HMT). *Procedia Engineering, 29*, 3017-3022.

Zollmann, A., & Venugopal, A. (2006). Syntax augmented machine translation via chart parsing. *In Proceedings of the Workshop on Statistical Machine Translation* (pp. 138-141). New York City, USA: Association for Computational Linguistics.