Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(209) 27: 437 – 452
© TÜBİTAK
doi:10.3906/elk-1807-341

A tree-based approach for English-to-Turkish translation

Özge BAKAY[1], **Begüm AVAR**[2], **Olcay Taner YILDIZ**[3,*]
[1]Department of Foreign Language Education, Boğaziçi University, İstanbul, Turkey
[2]Department of Linguistics, Boğaziçi University, İstanbul, Turkey
[3]Department of Computer Engineering, Faculty of Engineering, Işık University, İstanbul, Turkey

Received: 31.07.2018 • **Accepted/Published Online:** 29.11.2018 • **Final Version:** 22.01.2019

Abstract: In this paper, we present our English-to-Turkish translation methodology, which adopts a tree-based approach. Our approach relies on tree analysis and the application of structural modification rules to get the target side (Turkish) trees from source side (English) ones. We also use morphological analysis to get candidate root words and apply tree-based rules to obtain the agglutinated target words. Compared to earlier work on English-to-Turkish translation using phrase-based models, we have been able to obtain higher BLEU scores in our current study. Our syntactic subtree permutation strategy, combined with a word replacement algorithm, provides a 67% relative improvement from a baseline 12.8 to 21.4 BLEU, all averaged over 10-fold cross-validation. As future work, improvements in choosing the correct senses and structural rules are needed.

1. Introduction

Machine translation has been improving substantially over the last few years. The recent developments in machine translation are considered to be mainly obtained thanks to the increased computational power and the parallel corpora. It is also observed that current approaches in machine translation have been shifting from string/word-based models towards tree/phrase-based models and, in recent years, towards deep learning/neural network-based models.

As opposed to word-based models, phrase and tree-based models are found to be more efficient in translating from analytic languages, e.g., English, to agglutinative languages, e.g., Turkish. In phrase-based models, the main unit to be translated is the phrase. Some works, such as [1–5], apply this approach to translate from English to Turkish. However, a disadvantage of this approach is that as the number of words intervening between the related words increases, the phrase-based approach fails to link those related words. On the contrary, tree-based approaches, which use syntactic structure information in translation, are capable of considering the whole sentence at once and are therefore better at capturing the recursive structure of language as opposed to phrase-based models [6].

In this paper, we propose a tree-based approach for English-to-Turkish translation. Our contributions are twofold: (i) we continue the manual translation approach given in [7] and translate an extra 4K sentences from Penn-Treebank [8], and (ii) we propose a novel tree-based translation approach. Our tree-based translation approach has three phases. In the first phase, we translate a set of preselected words to *NONE* according to their POS tags. In the second phase, for any node in the tree, we make a decision based on the rules to either

[*] Correspondence: olcaytaner@isikun.edu.tr

permute its children or leave them as they are. In the last phase, we choose a root word for each nontranslated English word and agglutinate it with a set of morphemes to obtain its Turkish translation.

This paper is organized as follows: in Section 2, we give a brief review of the relevant structures in Turkish. We give a review of machine translation and its application on English-Turkish language pairs in Section 3. We give the details of our parallel corpus and present how it is constructed in Section 4. We explain our method in Section 5 and give the results of our experiment in Section 6. Lastly, we conclude in Section 7.

2. Turkish

In linguistics, the term morphology refers to the study of the internal structure of words. Each word is assumed to consist of one or more morphemes, which can be defined as the smallest linguistic unit having a particular meaning or grammatical function. One can come across morphologically simple words, i.e. roots, as well as morphologically complex ones, such as compounds or affixed forms.

One of the central foci of the literature on Turkish linguistics has been the complexity of Turkish morphology. Turkish is a textbook example of an agglutinative language, i.e. words in their surface form may contain various morphemes, such as multiple suffixes attached to a single base, each of which have a semantic and/or syntactic contribution.

(1) Batı-lı-laş-tır-ıl-a-ma-yan-lar-dan-mış-ız
west-With-Make-Caus-Pass-Neg.Abil-Nom-Pl-Abl-Evid-A3Pl
‘It appears that we are among the ones that cannot be westernized.’

The morphemes that constitute a word combine in a (more or less) strict order. Most morphologically complex words have the structure ROOT-SUFFIX1-SUFFIX2-... Affixes have two types: (i) derivational affixes, which change the meaning and sometimes also the grammatical category of the base they are attached to, and (ii) inflectional affixes serving particular grammatical functions. In general, derivational suffixes precede inflectional ones. The order of derivational suffixes is reflected on the meaning of the derived form. For instance, consider the combination of the noun göz, ‘eye’, with two derivational suffixes -lIK and -CI: even though the same three morphemes are used, the meaning of a word like göz-cü-lük, ‘scouting’, is clearly different from that of göz-lük-çü, ‘optician’.

Turkish is typically treated as a strict-final subject-object-verb (SOV) language, but a central problem with Turkish is that it does not have strict word order. In other words, the constituents of a sentence can be scrambled such that they can occur in any order with slight modifications in the sentential meaning [9]. In Turkish, the problem of parsing is even more advanced, as not only content words in their base forms but also functional morphemes, such as affixes, may be a source of ambiguity that cannot be resolved without the context.

3. Machine translation

Although the idea of machine translation goes back to the ALPAC (Automatic Language Processing Advisory Committe) report of 1966 [10], the first systems were mainly built on selected domains, such as weather forecasting (METEO produced by Montreal University [11]). The main property of these initial systems is that they applied a new approach for their time, rule-based translation.

After these initial attempts, the literature moves on to more complex approaches, such as statistical machine translation. One of the most successful statistical machine translation systems is CANDIDE, developed by IBM [12]. These word-based models, especially IBM Model 5, have a distinct place in word-based statistical machine translation systems. Several approaches were applied to improve the accuracy of IBM models [13, 14]. Due to the agglutinative nature of Turkish, for English-Turkish language pairs, it is not possible to obtain a 1-1 correspondence. Usually, English words are matched with the suffixes of Turkish words. For example, the English sentence "Tomorrow I will go to Ankara" will be translated into Turkish as "Yarın Ankara'ya gideceğim". Here "Tomorrow" matches with one word, "Yarın", whereas 3 words in English, "I will go", match with one word in Turkish, "gideceğim", and two words, "to Ankara", match with one word, "Ankara'ya" [1].

Phrase-based translation approaches attempt to overcome these difficulties. In these approaches, the main unit of translation is a phrase, which is obtained by grouping a set of consecutive words or suffixes. Since consecutive words are ordered, the sentence fluency is much better than that of word-based models [15, 16].

Most of the successful approaches in English-Turkish translation follow this approach [5]. The most crucial property of these approaches is their ability of word-to-word, word-to-morpheme, and morpheme-to-morpheme alignments. However, in order to have a significant improvement in translation, one has to preprocess a paramount portion of the corpus [2, 3]. The aim of this preprocessing is to allow word- and morpheme-based alignments and to solve the data sparsity problem for Turkish, which is a highly agglutinative language with an almost infinite number of possible word forms [17]. As an example of preprocessing, {I, will, go} and {to, Ankara} will be aligned manually with the morphemes {-(I)m, -(y)AcAk, git-} and {-(y)A, Ankara}, respectively. Here, "-(I)m" and "-(y)AcAk" are examples of morphemes and they correspond to morphological categories.

Although they are successful, neither word- nor phrase-based approaches can reflect the recursive structure of the language in translation, which decreases the syntactic performance of the translation. Both word- and phrase-based translation systems see the sentences as an ordered set of words and omit the syntactic connections between long-distance words, also called long-distance relationships. To overcome this problem, new strategies integrate the tree syntactic tree structure into translation [6].

In tree-based translation, the system searches for the alignments of words and syntactic forms stored in tree structure [18, 19]. After extracting the rule patterns from these alignments, the system estimates the probability distributions of these patterns and generates a statistical translation model. As a last step of translation, one decodes the input sentence. In this phase, several chart-parsing algorithms come in handy. These techniques start from an initial set of phrases and generate a chart of subtrees by using dynamic programming. The most probable subtrees are combined to produce bigger subtrees and, at the end, the translated tree. Hierarchical phrase-based model Hiero [20] follows this approach and does not use any explicit annotation.

The only tree-translation approach for English-Turkish pair is given by Gorgun et al. [7]. There are three important steps in their approach. First, for child permutation, they learn the probabilities for each production rule. At the end of training, each production rule is associated with a set of permutations and their counts. The counts are then normalized to probabilities. Second, for each English word w_e in a leaf of the tree, they learn a leaf probability distribution based on the previous translations of w_e. For this case, they generate not only probabilities for replacements between whole words, but also probabilities for replacing English function words with their Turkish morphemes counterparts. Third, in the testing phase, they combine the child permutation probabilities with the leaf replacement ones to calculate the probability of each translation candidate. The best translation is obtained by using the N-best lists.

4. Constructing a parallel corpus

An English-Turkish parallel treebank [7] was aligned from a subset of the original Penn-Treebank II corpus [8]. First, they permuted the subtrees in accordance with the Turkish sentence structure rules. Then leaf tokens were replaced with the most synonymous Turkish counterparts. In our work, we follow the same approach as in [7] to manually convert English parse trees to equivalent Turkish ones. The details of our approach are explained below.

In converting an English parse tree to its equivalent in Turkish, we follow two basic rules. The first rule is that we replace the English word at a leaf node with one or more Turkish words (at most 3 words). Conversely, more than one word in English may correspond to a single word in Turkish. In that case, we replace some English words with *NONE*. As an example of leaf replacement, when translating the sentence in Figure 1a, "Public" and "Affairs" in English are replaced with "Kamu" and "İşleri" in Turkish (Figure 1b).

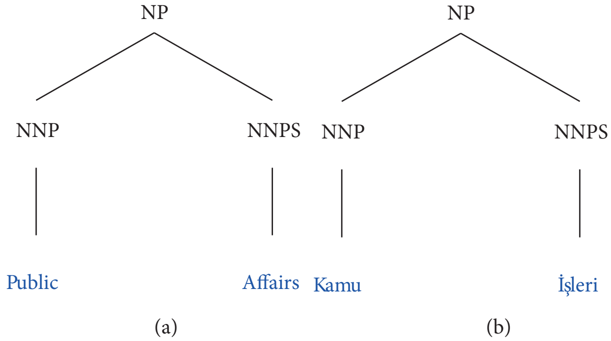


Figure 1. The English noun phrase "Public Affairs" translated into the Turkish noun phrase "Kamu İşleri".

With the second rule, we permute the children of one more node to reflect the syntactic shift from the English side to the Turkish side. For example, the majority of Turkish sentences have the Subject-Object-Verb word order whereas most English sentences have the Subject-Verb-Object order. As an example, when translating the sentence in Figure 2a, VBZ and NP subtrees are permuted so that the correct constituent order in Turkish is constructed in the translated form (Figure 2b).

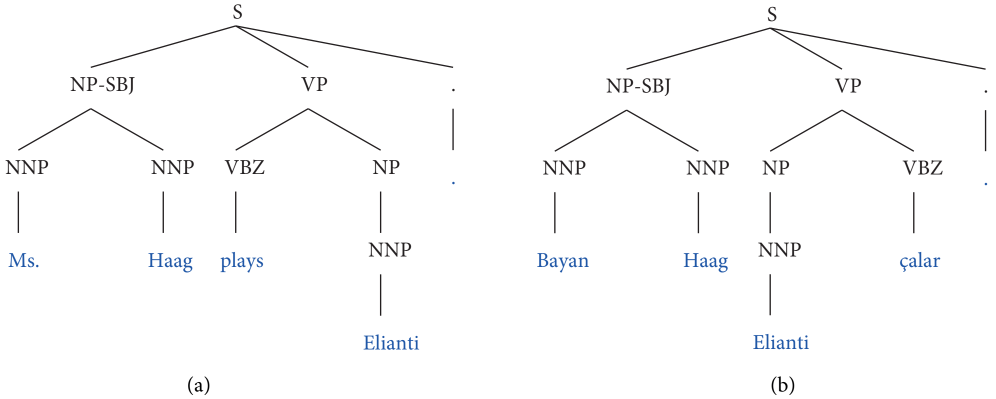


Figure 2. The English sentence "Ms. Haag plays Elianti" translated into the Turkish sentence "Bayan Haag Elianti çalar".

The semantic information expressed by modals and verb tenses in English, in general, corresponds to specific morphemes attached to the corresponding word stem in Turkish. For example, when translating the sentence in Figure 3a, the subtree containing the passive voice “hasn’t been set” (denoted by the uppermost VP node) is translated into a single word, “belirlenmedi”. In this case, the POS tag sequence of VBZ-RB-VBN-VBN in the English sentence is permuted to get the POS tag sequence VBN-VBN-RB-VBZ in the Turkish sentence, which also corresponds to the morphological analysis of “belirlenmedi” (belirle: to set, -(I)n: passive, -mA: negation, -DI: past) (Figure 3b). Note that the *NONE* tag is used to replace “has”, “n’t”, and “been” words since there is no direct word in Turkish for these English tokens.

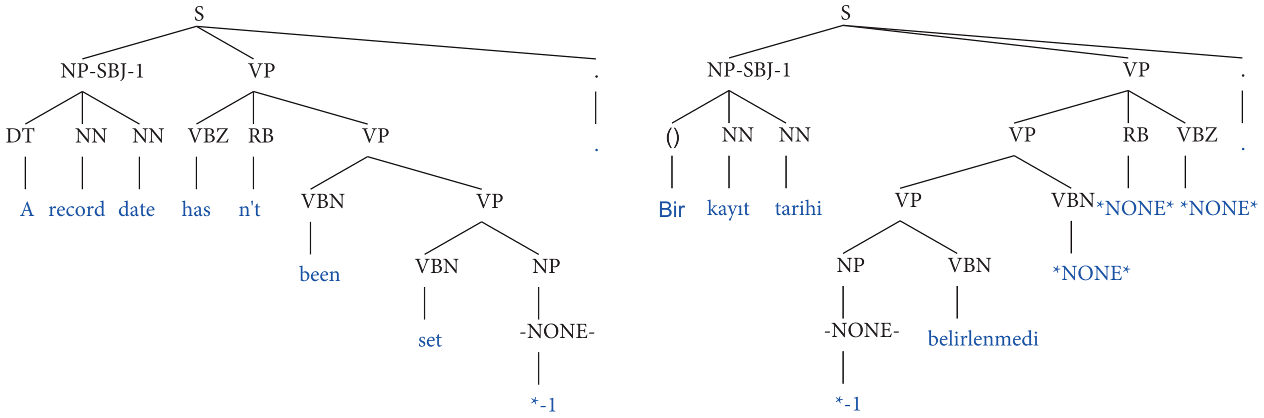
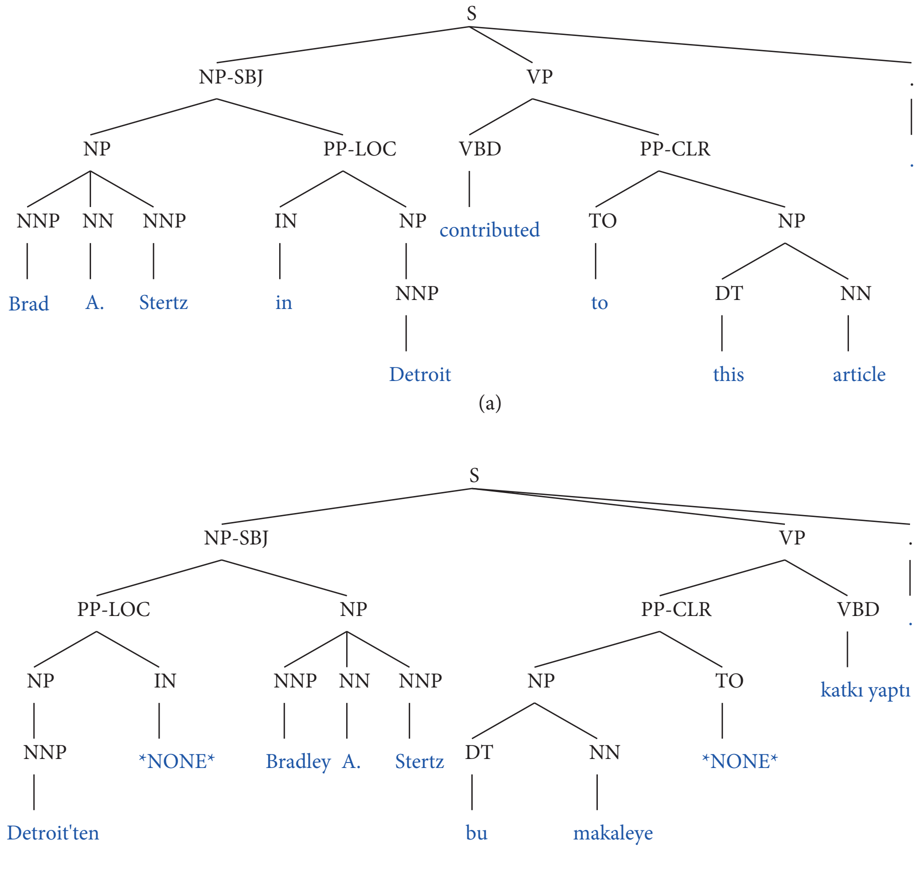


Figure 3. The English sentence “A record date hasn’t been set” translated into the Turkish sentence “Bir kayıt tarihi belirlenmedi”.

Similar cases occur in translation of subsentences with prepositions. For example, when translating the sentence in Figure 4a, the subtrees containing the prepositional phrases “in Detroit” (denoted by the PP-LOC node) and “to this article” (denoted by the PP-CLR node) are translated into phrases of “Detroit’ten” (-DAn: ablative) and “bu makaleye’’ (bu: this, makale: article, -(y)A: dative), respectively. The following changes can be observed (Figure 4b):

- The POS tag sequence of IN-NNP on the English side is permuted to get the POS tag sequence NNP-IN on the Turkish side, which also corresponds to the morphological analysis of “Detroit + -DAn”.
- The POS tag sequence of TO-NP on the English side is permuted to get the POS tag sequence of NP-TO on the Turkish side, which also corresponds to the morphological analysis of “makale + -(y)A”.
- The single word “contributed” on the English side is replaced with the multiword expression “katkı yap + DI” (katkı yap-: to contribute, -DI: past) on the Turkish side.
- The *NONE* tag is used to replace “to” and “in” words.

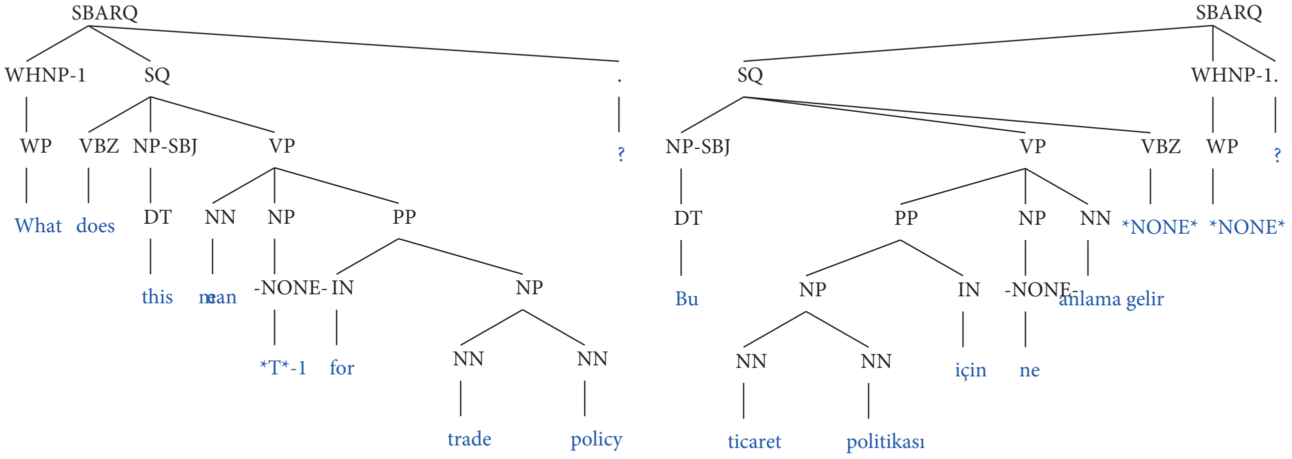
In the Penn-Treebank II annotation, the movement in English question sentences leaves a trace and is associated with a wh- constituent. Both the trace and the wh- constituent are indicated by the same numeric marker. For example, “WHNP-1” and “*T*-1” are associated. When we translate a sentence tree of a question, we replace the wh- constituent with *NONE* and replace its trace with the appropriate question pronoun in Turkish since Turkish is a wh-in-situ language. For example, when translating the sentence in Figure 5a, the “What” word is replaced with the *NONE* null token, whereas its associated marker “*T*-1” is replaced with the corresponding translation “ne” (Figure 5b).



(a)

(b)

Figure 4. The English sentence “Bradley A. Stertz in Detroit contributed to this article” translated into the Turkish sentence “Detroit’ten Bradley A. Stertz bu makaleye katkı yaptı”.



(a)

(b)

Figure 5. The English question sentence “What does this mean for trade policy?” translated into the Turkish question sentence “Bu ticaret politikası için ne anlama gelir?”.

5. Method

Our translation approach tries to mimic the human translation approach given in Section 4. To accomplish this aim, we follow the algorithm below:

1. We leave the punctuation symbols as they are.

2. We leave the null elements in the English tree as they are. Either the null elements contain "*" or their parents have "-NONE-" in their texts. For example, the node with text "*-1" in Figure 3 is a null element [8].

3. We translate a set of preselected words to *NONE* according to their POS tags. See Section 5.1 for details.

4. For each node in the tree, we permute its children or leave them as they are. The decision is based on a set of predefined rules. See Section 5.2 for details.

5. For each nontranslated English word w_e in the tree,

   (a) we choose a Turkish root word w_t depending on the previous translations of w_e (see Section 5.3 for details),

   (b) we find the correct set of morphemes $m_1, m_2 \dots$ (depending on the words replaced with *NONE* and their POS tags) and concatenate them (according to the Turkish morphotactic rules[1] [21]) with w_t to get the Turkish translation $w_t m_1 m_2 \dots$ of the word w_e. See Section 5.4 for the details of the selection of morphemes.

Figure 6 shows an example translation (sentence in Figure 3a) step by step. In Step 1, only the punctuation characters are left as they are. In Step 2, we leave null elements as they are. Thus, there is actually no change at the end of Step 2 (Figure 6a). In Step 3, we translate words with VBZ, RB, and VBN to *NONE* to get the tree in Figure 6b. In Step 4, we permute *NONE* subtrees and get the resulting tree in Figure 6c. As the last step, in Step 5, for each nontranslated English word ('A', 'record', 'date', ''set'), we choose the Turkish root words ('bir', 'kayıt', 'tarih', and 'belirle', respectively) depending on the previous translations of these words in the trees of the training set. We also find the correct set of morphemes for VBZ, RB, and VBN ('-(I)n', 'mA', and '-DI', respectively). If we concatenate these morphemes with the root word 'belirle', we get the resulting translated tree in Figure 3b.

5.1. *NONE* Replacement

While translating from English to Turkish, some words on the English side are not aligned but actually transformed into morphemes in the equivalent Turkish words ("has", "n't", "been" in Figure 3 and "to", "in" in Figure 4). For this reason, for tree translation, we need to replace those words with null tokens, namely with *NONE* in this context (see Table 1 for a detailed list).

[1] https://github.com/olcaytaner/MorphologicalAnalysis

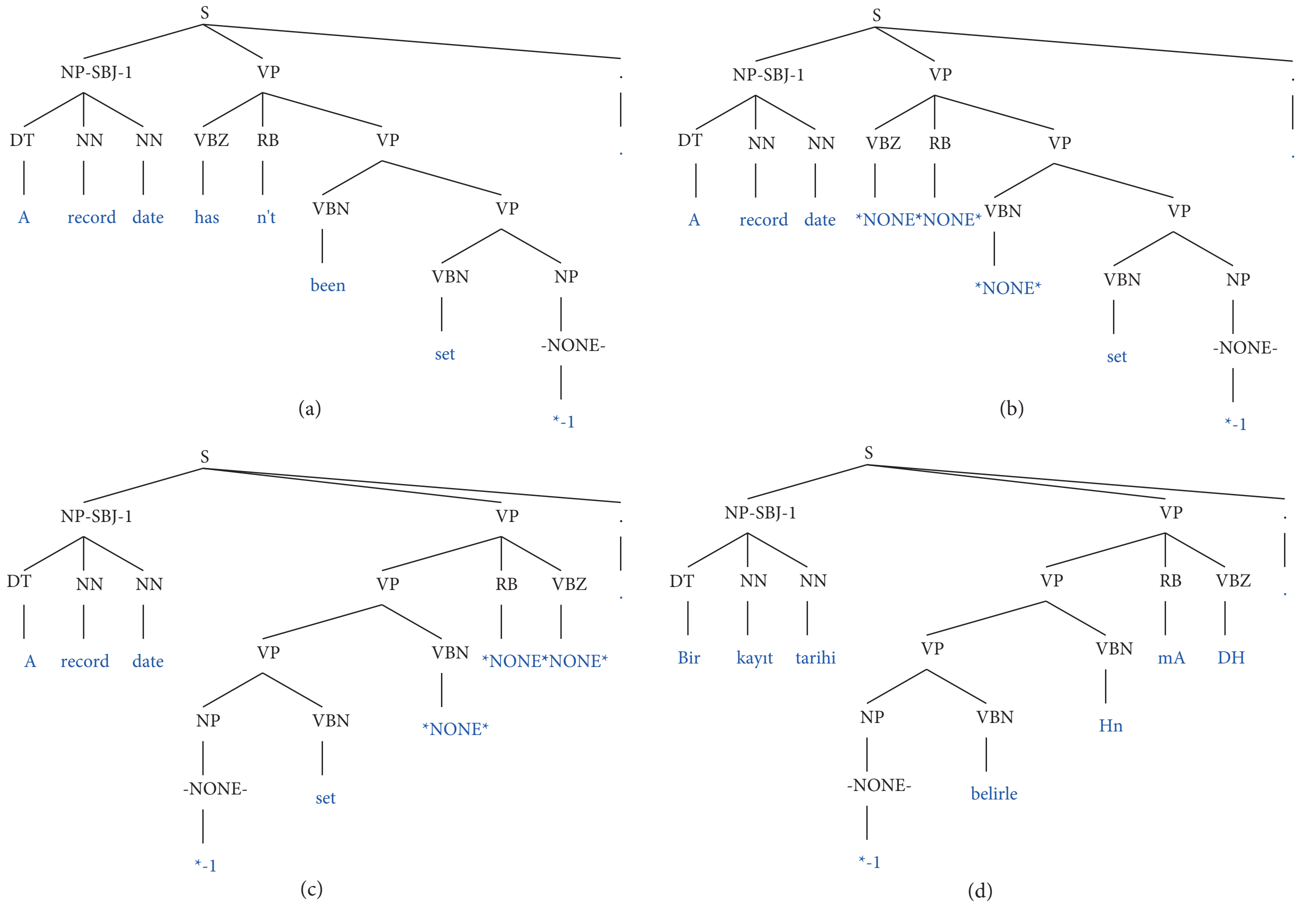


Figure 6. Steps in the proposed translation algorithm for the English sentence “A record date hasn’t been set.”

Table 1. Words to be replaced with null token *NONE*.

POS	Word(s)
DT	the
RB	-n’t, not
VB	have, be
POS	’s, ’
VBZ	has, does, is, ’s
VBN	been
IN	in, than, on, from, with, of, at, if, by
VBP	’re, is, are, am, ’m, do, have, has, ’ve
WP	who, where, which, what, why
TO	to
VBD	had, did, were, was
MD	will, ’d, ’ll, ca, can, could, would, should, wo, may, might

5.2. Subtree permutation

5.2.1. *NONE* subtree permutation

After replacing the words in Table 1 with the *NONE* token, we need to permute the parents of those nodes so that their translated morphemes can be agglutinated to the candidate Turkish root words. Let a parent node n_e have child nodes $c_1 \ldots c_{i-1} c_i \ldots c_j c_{j+1} \ldots c_k$ where the descendants of the child nodes $c_i \ldots c_j$ have been replaced with *NONE* tokens; to translate n_e, $c_i \ldots c_j$ are reversed and put at the end of the child list. Thus, the translated parent node n_t will have the child nodes $c_1 \ldots c_{i-1} c_{j+1} \ldots c_k c_j \ldots c_i$. For example, in Figure 3, the child nodes of VBZ and RB of the uppermost VP node are reversed and put at the end of VP's child list. Also, in Figure 4, the child nodes of IN and TO are put at the end of their parents', PP-LOC and PP-CLR's, child lists, respectively.

5.2.2. Syntactic subtree permutation

Other than *NONE* subtree permutations, we also need to permute other subtrees to reflect the syntactic changes on the Turkish side. Although there are many variants of each permutation possible, we select the most frequent syntactic permutations and apply them. Table 2 shows the top 5 applied syntactic subtree permutation rules.

Table 2. Top 5 applied syntactic subtree permutations.

Rule	Transformed rule	English	Turkish
NP → NP PP	NP → PP NP	(language) (in the law)	(yasadaki) (dil)
ADJP → JJ PP	ADJP → PP JJ	(sharp) (as a knife)	(bıçak gibi) (keskin)
NP → DT JJ NN	NP → JJ DT NN	(a) (sharp) (knife)	(keskin) (bir) (bıçak)
NP → NNP CD	NNP → CD NNP	(Oct.) (31)	(31) (Ekim)
VP → VBD NP PP	VP → PP NP VBD	(threw) (the book) (to the table)	(masaya) (kitabı) (attı)

5.3. Root word selection

After tree permutations, the tree structure is fixed and the next step is to find the Turkish counterparts of the English words in the leaf nodes. At this stage of the algorithm, we use a pretranslated parallel corpus, which is obtained by using the techniques explained in Section 4. For each English word, we store the glosses translated by the human translators in the training set. The list of glosses is sorted according to their likelihood. For this part of the algorithm, as the size of the corpus grows, we expect an increase in the performance of the algorithm.

Table 3. Top 15 possible translations of the word "industry".

Word	Count	Word	Count	Word	Count
endüstri	46	endüstrisinde	8	endüstriden	3
endüstrisi	25	endüstride	7	endüstriye	2
Endüstri	15	endüstriyi	6	Endüstrideki	2
endüstrisinin	13	sanayi	5	sanayinin	2
endüstrinin	11	endüstrisine	5	endüstridekileri	1

Table 3 shows the top 15 possible translations of the word "industry". As we can see, although there are two possible translations of word "industry" as "endüstri" and "sanayi", due to the highly agglutinative nature of Turkish, many possible translations occur in the dataset. To overcome this problem, we morphologically analyze and disambiguate each possible translation, get a total sum for each possible root word, and select the root word with the maximum sum as the candidate root word. For the case in Table 3, root words "endüstri" and "sanayi" have sums of 144 (46 + 25 + 15 + 13 + 11 + 8 + 7 + 6 + 5 + 3 + 2 + 2 + 1) and 7 (5 + 2), respectively.

Note that, if there are multiple Turkish words for a corresponding English word, we only analyze the last Turkish word. For example, "contributed" has two main translations, "katkıda bulun+..." and "katkı yap+...". For those cases, we only analyze "bulun+..." and "yap+..."

Note also that, while doing morphological analysis, we only consider the root words whose POS tags match with the ones of the corresponding English words.

If there are no translated glosses for an English word, we search an English-Turkish dictionary for this word to obtain a candidate translation. If the word does not exist in the bilingual diction, either, we simply do not translate the word, i.e. leave the word as it is.

5.4. Agglutination

After the root word selection, we need to decide the morphemes to be added to the root word. Subtree permutations helped us by placing *NONE* tokens after each root word in the tree. Thus depending on the *NONE* tokens, corresponding English words, and the POS tags of those words, we can construct the translation of the agglutinated Turkish word. In this section, we define a *NONE* tuple as (x, y), where x is the corresponding English word and y its POS tag. According to this definition:

- In Figure 3, we select the word "belirle" (to set) as the root word. The next three *NONE* tuples are (been, VBN), (n't, RB), and (has, VBZ). For these three tuples, we select -(I)n, -mA, and -DI as morphemes, respectively. When we concatenate these morphemes, we get the Turkish word "belirle-n-me-di".

- In Figure 4, we select the words "Detroit" and "makale" (article) as the root words. The next *NONE* tuples are (in, IN) and (to, TO), respectively. For these tuples, we select -DAn and -(y)A as morphemes, respectively. When concatenated, we get the words "Detroit-ten" and "makale-ye'', respectively.

- In Figure 5, we select the word "anlama gel" (to mean) as the root word. The next two *NONE* tuples are (does, VBZ) and (what, WP). We only select -(I)r as a morpheme. When concatenated, we get the word "anlamına gelir".

5.4.1. Nouns

The noun category includes NN, NNP, NNS, and NNPS tags. NNPS and NNP tags are treated similarly and the rules for them are nearly the same as the rules for the NN tag, only with ' added in front of the morpheme. For nouns, we only check the next *NONE* tuple, and depending on that tuple, we decide on the morpheme to be concatenated with the root word. The morphemes associated with *NONE* tuples for NN and NNP tags are given in Tables 4 and 5, respectively.

Table 4. Corresponding Turkish morphemes of *NONE* tuples for the NN tag.

NONE tuple(s)	Morpheme
('s, VBZ), (is, VBZ)	-DIr
(are, VBP), ('re, VBP)	-DIr
(in, OF), (, POS)	-(n)In
(in, IN), (on, IN), (at, IN)	-DA
(than, IN), (from, IN), (since, IN)	-DAn
(his, PRP), (her, PRP), (its, PRP)	-(s)I
(our, PRP)	-(I)mIz
(into, IN), (until, IN)	-(n)A
(with, IN), (by, IN)	-(y)lA
(to, TO)	-(y)A
(my, PRP)	-(I)m
(your, PRP)	-(I)n
(their, PRP)	-lArI

Table 5. Corresponding Turkish morphemes of *NONE* tuples for the NNP tag.

NONE tuple(s)	Morpheme
('s, VBZ), (is, VBZ)	-DIr
(are, VBP), ('re, VBP)	-DIr
(in, OF),	-(n)In
(in, IN), (on, IN), (at, IN)	-DA
(into, IN)	-(n)A
(with, IN), (by, IN)	-(y)lA
(to, TO)	-(y)A
(from, IN), (since, IN)	-DAn

5.4.2. Adjectives

The adjective category includes JJ, JJR, and JJS tags. The same as for nouns, we examine the next *NONE* tuple and decide on the morpheme depending on that tuple. The behavior of the system is similar for all the three tags, with two exceptions: before the candidate root word, (i) for the JJR tag, we add the word "daha" (more), and (ii) for the JJS tag, we add the word "en" (the most). The morphemes associated with *NONE* tuples for those tags are given in Table 6.

Table 6. Corresponding Turkish morphemes of *NONE* tuples for the JJ tag.

NONE tuple(s)	Morpheme
('s, VBZ), (is, VBZ), (are, VBP), ('re, VBP)	-DIr
(was, VBN), (were, VBN)	-(y)DI

5.4.3. Numerals

The numeral category includes the CD tag. The same as four nouns, we examine the next *NONE* tuple and decide on the morpheme depending on that tuple. There are two categories in the numeral translation. The numerals are either a text such as "million" or "two" or they contain digits such as 123 or 12.45. For text numerals, in English-Turkish translation, the morphemes are concatenated as they are, such as "milyon-a" (milyon: million, -(y)A: dative) or "iki-den" (iki: two, -DAn: ablative). If the numeral contains only digits, the morphemes are preceded by an apostrophe like "123'e" (-(y)A: dative) or "12.45'ten" (-DAn: ablative). The morphemes associated with *NONE* tuples for CD tags are given in Table 7.

Table 7. Corresponding Turkish morphemes of *NONE* tuples for the CD tag.

NONE tuple	Morpheme
(in, IN)	-DA
(to, TO)	-(y)A
(from, IN), (than, IN)	-DAn

5.4.4. Verbs

Verbs have the most complex morphotactics in both English and Turkish. Usually, in order to obtain the Turkish translation of an English verb, we concatenate more than one morpheme. Thus, for the verb tags, we need not only the next *NONE* tuple, but more than one following *NONE* tuple. The added morphemes associated with *NONE* tuples for VB, VBG, and VBN tags are given in Tables 8, 9, and 10, respectively. For VBZ and VBP tags, we concatenate the morpheme -(I)r or -(A)r depending on the number of the syllables of the Turkish root verb. For the VBD tag, we add only the morpheme -DI.

Table 8. Corresponding Turkish morphemes of *NONE* tuples for the VB tag.

NONE tuple(s)	Morpheme
(will, MD)	-(y)AcAk
(will, MD)+(, RB)	-mA + (y)AcAk
(can, MD), (may, MD), (might, MD), (could, MD)	-(y)Abil + (I)r
(can, MD)+(, RB), (may, MD)+(, RB), (might, MD)+(, RB), (could, MD)+(, RB)	-mAz
(would, MD), (wo, MD)	-(I)yor
(would, MD)+(, RB), (wo, MD)+(, RB)	-m + (I)yor
(to, TO)+(have, VB)	-mAlI
(to, TO)+(had, VBD)	-mAlI + (y)DI
(, RB)+(did, VBD)	-mA + DI
(, RB)+(do, VBP), (, RB)+(does, VBZ)	-mAz

6. Experiments

6.1. Setup

We manually convert 4K English parse trees to equivalent Turkish ones. Our converted treebank including the previous trees of [7] covers 9559 sentences with a maximum length of 15 tokens, including punctuation. The experimental setup for English-Turkish translation is as follows:

Table 9. Corresponding Turkish morphemes of *NONE* tuples for the VBG tag.

NONE tuple(s)	Morpheme
(, VB)+(, MD)	-(y)AcAk
(, RB)+(, VB)+(, MD)	-mA + (y)AcAk
(, VBZ), (, VBP), (, VBN)	-(I)yor
(, RB)+(, VBZ), (, RB)+(, VBP), (, RB)+(, VBN)	-m + (I)yor
(, VBD)	-(I)yor + DI
(, RB)+(, VBD)	-m + (I)yor + DI

Table 10. Corresponding Turkish morphemes of *NONE* tuples for VBN tag.

NONE tuple(s)	Morpheme
(is, VBZ), ('s, VBZ), (are, VBP), ('re, VBP)	-(I)r
(, RB)+(is, VBZ), (, RB)+('s, VBZ), (, RB)+(are, VBP), (, RB)+('re, VBP)	-mAz
(be, VB)+(can, MD), (be, VB)+(will, MD)	-(y)Abil + (I)r
(be, VB)+(, RB)+(can, MD), (be, VB)+(, RB)+(will, MD)	-(y)A + mAz
(be, VB)+(could, MD), (be, VB)+(would, MD)	-(y)Abil + (I)r + DI
(be, VB)+(, RB)+(could, MD), (be, VB)+(, RB)+(would, MD)	-(y)A+ mAz + DI
(was, VBD), (were, VBD), (been, VBN)+(has, VBZ)	-DI
(, RB)+(was, VBD), (, RB)+(were, VBD), (been, VBN)+(, RB)+(has, VBZ)	-mA + DI
(been, VBN)+(had, VBD)	-(I)yor + DI
(been, VBN)+(, RB)+(had, VBD)	-m +(I)yor + DI
(has, VBZ), ('s, VBZ), (have, VBP), ('ve, VBP), (had, VBD)	-mIş + DI
(, RB)+(has, VBZ), (, RB)+('s, VBZ), (, RB)+(have, VBP), (, RB)+('ve, VBP)	-mA + DI
(, RB)+(had, VBD)	-mA +mIş + DI

- For obtaining training and testing sets, we do 10-fold cross-validation. At each fold, we use 90% of the parse trees for training (8604 trees at each fold) and 10% for testing (956 trees at each fold). We report the average and the standard deviation of the results over those 10 folds.

- At each fold, we use the parse trees in the training set for the root word selection (see Section 5.3). We compare our results of automatic translation with that of human translation for the test data.

- We use the Oxford English-Turkish bilingual dictionary for out of vocabulary cases (see Section 5.3).

6.2. Translation quality

First, we compare our tree-translation approach with the previous tree-translation approach of Gorgun et al. [7] on English-Turkish language pairs. We also develop two baseline systems representing structure learning and word replacement. In the first baseline system, we replace English words with their most frequent Turkish translations based on a pretranslated parallel corpus. Since we do not change the structure of the tree in this baseline, we call it baseline (original structure). In the second baseline system, we do both syntactic subtree permutation as explained in Section 5.2.2 and word replacement as in the first baseline. We call this baseline (learned structure).

Table 11 shows the BLEU [22] scores for all the five tree-based algorithms. First, the baseline system based on the original English tree structure performs similarly to the learned-model of Gorgun et al. Second, the structure transformation strategy gives us improvement of 43% over the original tree structure (from 12.8 to 18.3). Third, simple most-frequent-word-replacement is not enough. If we apply the agglutination rules explained in Section 5.4, we get another 24% improvement over the first baseline system (from 18.3 to 21.4).

Compared to the previous tree approaches, our results (21.4 ± 1.0) are much better than the previous tree approach adopted by Gorgun et al. where they obtained a BLEU score of 12.8 for the learned-tree setup (they learn the tree structure from the training data) and 16.3 for the optimal tree set-up (they assume that the corresponding Turkish tree structure of a translation is given). The main reason for this difference is the rule-based vs. statistical dilemma. First, our approach takes the Occam's razor approach and uses few built-in syntactic subtree permutation rules, whereas Gorgun et al. try to learn those rules and, due to overfitting, may be applying the wrong permutation rules at the subtrees. Second, although Gorgun et al. generate a probability distribution over leaf replacement and use this distribution in searching for the optimal tree, our approach of replacing leaves with the most occurring root words combined with built-in agglutination rules for morphemes is both simpler and better. In this case, due to the high search complexity, N-best lists may not find the correct morphemes to be concatenated and this will result in a different and incorrect translation.

Yet another thing to note is the time complexity of the translation. Although not as important as the quality of the translation, translation time can also be an issue. Our approach is rule-based and it does not involve any search; therefore, if compared to the search-based approach of Gorgun et al., it is much faster.

Note that any mistranslation in the morphemes of a word results in the wrong translation of the whole word, which, in turn, decreases BLEU scores. For example, when a word is translated into "ev-e" (ev: house, -(y)A: dative) instead of "ev-i" (-(y)I: accusative), the whole word is taken as mistranslated due to the wrong case marker.

Table 11. BLEU scores for three tree-based algorithms.

Approach	BLEU Score
Baseline (original structure)	12.8 ± 0.5
Baseline (learned structure)	18.3 ± 0.4
Our algorithm	21.4 ± 1.0
Gorgun et al. (learned structure)	12.8 ± 0.5
Gorgun et al. (optimal structure)	16.3 ± 0.5

6.3. Structure agreement

In the second phase of our experiments, we compare the performance of our subtree permutation algorithm. To accomplish this aim, we compare our translated tree with the correct tree node by node. For each comparison, we check if the subtree permutations are the same or not. According to the experiment results, there is 81.93 ± 0.5% structure agreement between the correct translation and our translation. A couple of important factors to consider to improve the translation would be to include punctuation, to have the system learn the rules on its own based on the data it comes across, and to provide the system with a list of exceptions to the rules.

6.4. *NONE* replacement

In the third phase of our experiments, we check if we have correctly replaced null tokens. We compare each leaf node in our translated tree with its correspondent in the correct tree. We call a leaf node a true positive

(TP) if it has the null token both in our tree and the correct tree, a true negative (TN) if it does not have a null token both in our tree and the correct tree, a false positive (FP) if it has a null token in our tree but not in the correct tree, and a false negative (FN) if it does not have a null token in our tree but has a null token in the correct tree.

According to the experiments, the TP rate of our algorithm is 91.50 ± 0.51, whereas the FP rate is 4.71 ± 0.26. The high TP rate and low FP rate results show that we successfully determine *NONE* tokens and we rarely mislabel normal words as *NONE* tokens. To improve the statistics of *NONE* replacement in future work, we suggest learning the rules to label *NONE* constituents of the verb collocations ("up" should be replaced with *NONE* token for the collocation "get up") and differentiating the copula "be" and lexical "have" from their auxiliary counterparts and assigning them with the related senses.

7. Conclusion

In this paper, we have presented our current tree-based approach to English-to-Turkish translation. Given a constituency parse tree of an English sentence, we first replace the words that do not have their Turkish counterparts with the *NONE* token, and then use simple subtree permutation rules to transform this tree. Remaining untranslated leaf nodes are replaced with their maximum likelihood estimated Turkish counterparts. As a last step, depending on the *NONE* tokens, corresponding English words, and the POS tags of those words, we agglutinate selected morphemes with the appropriate root words to get a better translated Turkish words.

We show that our algorithm finds 80 percent of the subtree permutations and 90 percent of the *NONE* replacement correctly. We also show that, due to this high accuracy in those two important parts, our approach has better performance than the previous related study of Gorgun et al. [7] in terms of speed and quality. Although there are some similarities between these two competing approaches, as a general distinction, Gorgun et al.'s approach is statistical whereas our approach is rule-based in subtree permutation, *NONE* replacement, and morpheme selection for agglutination. The only statistical part in our algorithm is in root word selection, where we use maximum likelihood estimation for determining root words.

As a future work, we plan to improve this study in light of the shortcomings we have noticed. The points that need to be paid attention to in future work can be categorized into two: (i) sense annotation and (ii) structural improvements. As for the sense annotation, choosing the correct senses for the given words would help improve our current BLEU scores. An important factor in being able to choose the correct senses is to include the senses of collocations as well as those of single words. For example, in Figure 3a, to be able to obtain a better translation, the sense of the collocation of "record date" must be included in the lists of senses given for "record" and "date" individually.

Structure-wise, there are many points that need to be considered. To name a few, case markers required by some nouns, adjectives, adverbs, and verbs in Turkish need to be added. As was said in Section 4.1, lacking the translation of morphemes on their own is a big disadvantage for the BLEU scores as they cause the translation of the whole word to be counted as wrong. Additionally, nominalization of the verbs in embedded clauses, translation and coreference of reflexives to their antecedents, addition of the question particle "mI" in yes/no questions, and having adverbs, articles, embedded clauses, and determiners in their correct positions are among the other issues that need to be improved regarding morphology and syntax.

Acknowledgment

This work was supported by TÜBİTAK project 116E104.

References

[1] El-Kahlout ID, Oflazer K. Initial explorations in English to Turkish statistical machine translation. In: Workshop on Statistical Machine Translation; 2006; Stroudsburg, PA, USA. pp. 7-14.

[2] El-Kahlout ID, Oflazer K. Exploiting morphology and local word reordering in English-to-Turkish phrase-based statistical machine translation. IEEE T Audio Speech 2010; 18: 1313-1322.

[3] Oflazer K, El-Kahlout ID. Exploring different representational units in English-to-Turkish statistical machine translation. In: Workshop on Statistical Machine Translation; 2007; Prague, Czech Republic. pp. 25-32.

[4] Oflazer K. Statistical machine translation into a morphologically complex language. In: Intelligent Text Processing and Computational Linguistics; 2008; Haifa, Israel. pp. 376-387.

[5] Yeniterzi R, Oflazer K. Syntax-to-morphology mapping in factored phrase-based statistical machine translation from English to Turkish. In: 48th Annual Meeting of the Association for Computational Linguistics; 2010; Stroudsburg, PA, USA. pp. 454-464.

[6] Koehn P. Statistical Machine Translation. New York, NY, USA: Cambridge University Press, 2010.

[7] Gorgun O, Yildiz OT, Solak E, Ehsani R. English-Turkish parallel treebank with morphological annotations and its use in tree-based smt. In: International Conference on Pattern Recognition and Methods; 2016; Rome, Italy. pp. 510-516.

[8] Marcus MP, Marcinkiewicz MA, Santorini B. Building a large annotated corpus of English: The Penn treebank. Comput Linguist 1993; 19: 313-330.

[9] Erguvanlı ET. The Function of Word Order in Turkish Grammar. Berkeley, CA, USA: University of California Press, 1984.

[10] Hutchins J. ALPAC: The (In)famous Report. Cambridge, MA, USA: MIT Press, 2003.

[11] Thouin B. The METEO system. Practical Experience of Machine Translation. Amsterdam, the Netherlands: North Holland Publishing Co., 1982.

[12] Berger AL, Brown PF, Pietra SAD, Pietra VJD, Gillett JR, Lafferty JD, Mercer RL, Printz H, Ures L. The Candide system for machine translation. In: Workshop on Human Language Technology; 1994; New Jersey, USA. pp. 157-162.

[13] Och FJ, Ney H. A systematic comparison of various statistical alignment models. Comput Linguist 2003; 29: 1951.

[14] Vogel S, Ney H, Tillman C. HMM-based word alignment in statistical translation. In: 16th International Conference on Computational Linguistics; 1996; Copenhagen, Denmark. pp. 836-841.

[15] Och FJ, Tillmann C, Ney H. Improved alignment models for statistical machine translation. In: Joint Conference of Empirical Methods in Natural Language Processing and Very Large Corpora; 1999; Maryland, USA. pp. 20-28.

[16] Och FJ, Ney H. The alignment template approach to statistical machine translation. Comput Linguist 2004; 30: 418-449.

[17] Oflazer K. Turkish and its challenges for language processing. Language Resources and Evaluation 2014; 48: 639-653.

[18] Chiang D. A hierarchical phrase-based model for statistical machine translation. In: 43rd Annual Meeting of the Association for Computational Linguistics; 2005; Ann Arbor, MI, USA. pp. 263-270.

[19] Chiang D. Hierarchical phrase-based translation. Comput Linguist 2007; 33: 201-228.

[20] Chiang D, Lopez A, Madnani N, Monz C, Resnik P, Subotin M. The Hiero machine translation system: extensions, evaluation, and analysis. In: Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing; 2005; Vancouver, Canada. pp. 779-786.

[21] Gorgun O, Yildiz OT. A novel approach to morphological disambiguation for Turkish. In: Computer and Information Sciences II; 2011; Germany. pp. 77-83.

[22] Papineni K, Roukos S, Ward T, Zhu W. BLEU: A method for automatic evaluation of machine translation. In: 40th Annual Meeting of the Association for Computation Linguistics; 2002; Philadelphia, PA, USA. pp. 311-318.