JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 36, 309-322 (2020)
DOI: 10.6688/JISE.202003_36(2).0010

Unsupervised Weighting of Transfer Rules in Rule-Based Machine Translation using Maximum-Entropy Approach

SEVILAY BAYATLI[1], SEFER KURNAZ[1], ABOELHAMD ALI[2],
JONATHAN NORTH WASHINGTON[3] AND FRANCIS M. TYERS[4,5]
[1]*Department of Electrical and Computer Engineering*
Altınbaş Üniversitesi
Istanbul, 34217 Turkey
[2] *Department of Computer and Systems Engineering*
Alexandria University
Alexandria, 11432 Egypt
[3]*Linguistics Department, Swarthmore College*
Swarthmore, PA 19081 USA
[4]*School of Linguistics, Higher School of Economics*
Moscow, 101000 Russia
[5]*Department of Linguistics*
Indiana University
Bloomington, IN 47405 USA
E-mail: sewale.taha@ogr.atlinbas.edu.tr; sefer.kurnaz@altinbas.edu.tr;
aboelhamd.abotreka@gmail.ocm; jonathan.washington@swarthmore.edu; ftyers@iu.edu.tr

In this paper we present an unsupervised method for learning a model to distinguish between ambiguous se-lection of structural transfer rules in a rule-based machine translation (MT) system. In rule-based MT systems, transfer rules are the component responsible for converting source language morphological and syntactic structures to target language structures. These transfer rules function by matching a source language pattern of lexical items and applying a sequence of actions. There can, however, be more than one potential sequence of actions for each source language pattern. Our model consists of a set of maximum entropy (or logistic regression) classifiers, one trained for each source language pattern, which select the highest probability sequence of rules for a given sequence of patterns. We perform experiments on the Kazakh – Turkish language pair – a low-resource pair of morphologically-rich languages – and compare our model to two reference MT systems, a rule-based system where transfer rules are applied in a left-to-right longest match manner and to a state-of-the-art system based on the neural encoder–decoder architecture. Our system outforms both of these reference systems in three widely used metrics for machine translation evaluation.

Keywords: machine translation, weighting, structural transfer rules, ambiguous rules, disambiguation

1. INTRODUCTION

Machine translation (MT) is a procedure of translation from one language, the source language (SL), into second language, the target language (TL) through a computational model of translation via an intermediate representation (IR). These translation models may be differentiated based on both their knowledge source and intermediate representation. Rule-based, corpus-based, and hybrid approaches may be [1] based on

Received September 24, 2019, revised September 28 & October 11, 2019; accepted October 17, 2019.
Communicated by Osamah Ibrahim Khalaf.

types of knowledge used in their development.

The predominant approach to MT is corpus-based.[1] These approaches use large aggregations of parallel text (or bitexts) as the origin of knowledge. A parallel text is located beside its translation to learn a statistical model of translation. Creating these texts can be time-consuming and labour-intensive. However, if parallel text exists for a given a language pair in the arrangement of tens of millions of words, [2] such a system requires computational power but minimal direct human effort. In addition, by developing statistical or neural translation systems, we will ignore secondary language development, and barriers will be created between language groups. Digitalization should break these barriers, but current neural methods widen the gap between small groups of dominant languages and secondary ones.

Neural machine translation is considered a current approach for corpus-based machine translation which was first released by [3, 4], and was developed using sequence-to-sequence models. This approach encounters difficulty handling long sentences because the fixed-length vector representation does not contain sufficient competency to encode lengthy sentences with complicated structure and intent. Furthermore, the performance of NMT decreases rapidly as the number of unknown of words increases. This issue presents a challenge in increasing the magnitude of vocabularies employed by neural machine translation systems in the future. Another limitation of the NMT system is its incompatibility with the specific uses of certain organizations, thereby causing difficulties for them to refine and improve the system according to their needs.

An effective solution is to use hybrid methods that are based on combining the best features of two or more MT mechanisms [5]. Ehara [6] stated that the combination of rule-based and statistical methods had a positive effect on translation accuracy. We followed a similar path, which involves improving RBMT systems by using statistical MT (SMT) in our approach to the research problem.

In rule-based machine translation (RBMT) the translation process is based on using linguistic resources, such as computational morphological descriptions or grammar, bilingual dictionaries, and rules for disambiguation and structural transfer.

In this paper, we describe an extension to this system where we replace the left-to-right longest match algorithm with a search of possible rule combinations. In this context, we propose a novel unsupervised learning method in which shallow-transfer MT rules have been learned automatically from monolingual corpora by using an unsupervised maximum entropy approach. Thus, the annotated development corpus is not essential for calculation. The training procedure is based on obtaining translations for all possible combinations into the target language by using the rest of the modules in the MT pipeline, and then acquiring normalized probabilities for all translations from a language model to replace fractional counts in the supervised learning method. In this case, the performance of the target-language model can be surpassed by using only source-language information. The conflict between transfer rules is resolved by selecting the most suitable ones according to a global minimization function, rather than proceeding in a pairwise greedy manner. The reminder of the paper is as follows: Section 2 provides a brief review of previous research on Turkic language MT, Turkic and other language machine MTSs, and other recent MTSs. This section also provides an overview of other publicly accessible Kazakh-Turkish machine translators, and hybrid machine translation. Section 3 describes the system and tools used to build up weighted systems. Section 4 presents a

[1] This includes neural machine translation (NMT), where the intermediate representation is a vector representation of the sentence, and the classic "phrase-based" statistical machine translation, where the intermediate representation is typically the correspondence between fixed length sequences of words.

preliminary evaluation of the system. Section 5 discusses the results we obtained. Finally, Section 6 provides concluding remarks.

2. PREVIOUS WORK

RBMT systems have relied on handcrafted rules, which determine how a (syntactic or semantic) structure in a particular language fits into the corresponding structure in another language.

Within the scope of the Apertium [7] project, ongoing work builds underlying MT components (such as morphological transducer systems) to carry out translation between two Turkic languages, either between a Turkic language and Russian or a Turkic language and English and also between several other language pairs. There are released MT systems are for Kazakh-Turkish [8], Kazakh-Tatar [9], and English-Kazakh [10].

Statistical machine translation (SMT) emerged at the beginning of the 1990s and has registered many achievements in MT performance. These systems have strengths and weaknesses in providing the best translation. For this reason, the research community has switched its concentration toward integrating rule-based and statistical methods by combining the outputs of MT systems, which are known as hybrid systems. Studies have been conducted on constructing systems in which the statistical constituent is responsible for the translation and the second system supplies auxiliary information. In both cases, the results have been positive for out-of-domain testing. The other approach, in which the translation is led by the RBMT system and complementary information is provided by the SMT system, has been less explored. Habash *et al*. [11] improved the dictionary of an RBMT system with phrases from an SMT system. Results showed improvement to both systems, particularly hybrid systems translating into languages with more complex morphology than the source.

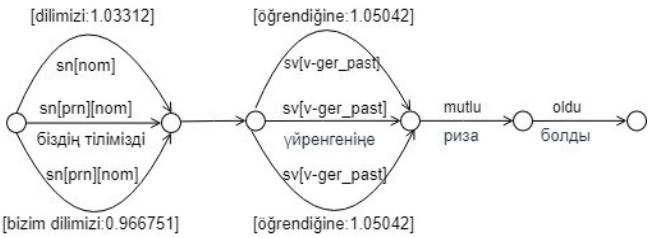


Fig. 1. The maximum-entropy rule-application process. The weights for each translation are summed and the translation with the highest associated weights is selected. For the words *біздің тіліміздi* ‘our language’ *үйренгеніңе* ‘to learn’, the translation *dilimizi öğrendiğine* is chosen, the translation of sentence *біздің тіліміздi үйренгеніңе риза болды* ‘she/he was happy to learn our language’ is *dilimizi öğrendiğine mutlu oldu.* 2.08354+1.05042=2.08354 is the highest weight over other translations.

3. SYSTEM

3.1 Methodology

According to the rule, lemma sequences and morphemes are translated one at a time.

This approach is not consistent, even for closely related languages. Structural transfer rules consist of two basic parts: namely, a pattern and an action, which are accountable for adapting the morphology or word arrangement to generate sufficient TL content.

The structural transfer module uses a sequence of SL and TL lexical form pairs in the following format: ^SL-lemma<SL-tag1><SL-tag2><…><SL-tagN>/TL-lemma<TL-tag1><TL-tag2><…><TL-tagN>$. TL lemma sequences and tags are supplied by the preceding two modules: lexical transfer and lexical selection. The lexical transfer module touches at the TL lemma and commonly the first one or two tags, and the rest of the tags are carried over from the SL.

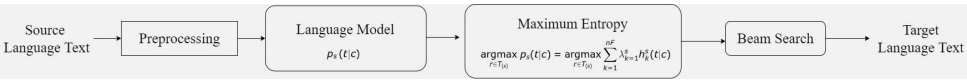


Fig. 2. Architecture of translation approach based on maximum entropy models.

These rules specify patterns (sequences) of source text lexical form and perform the coinciding transformations. Our approach has attempted to solve two main issues: First, the patterns were detected by the module in a left-to-right, longest-match way. For instance, the phrase *Бұл үлкен жетістік* 'This big success' was designated and handled by the rule for determiner-adjective-noun and not by the rule for determiner-adjective or determiner-noun because the first pattern is longer. Selecting the longest rule is not always correct, because for some sentences, the two shortest rules provide a better translation than that of the longest rule. Second, when two rules apply to the same pattern, the existing unweighted model selects the default rule, which is located first in the structure transfer rules file. This situation is considered problematic in selecting the correct rule for the most adequate translation. For instance, past-tense verbal adjectives have one form, 'V-GA$_H$' but can be translated as past-tense verbal adjectives or subject-relative verbal adjectives in Turkish. The first rule detects a pattern when the word being modified by the verbal adjective is the subject of the verbal adjective, which represents the form of the verb with the suffix -(y)An, and the second rule detects the pattern in which the word being modified by the verbal adjective has other functions, and represents a past-tense verbal adjective. The suffix -dik joins the verb and the subject of verbal adjectives, and is expressed by possessive agreement on the verbal adjective. For example, the Kazakh sentence *Сербия мен Қазақстан арасында алмас**қан** мәселе жоқ*, 'There are not any **unresolved** issues between Serbia and Kazakhstan' can be translated into Turkish as *Sırbistan ve Kazakistan arasında değiş**en** mesele yok*, and as *Sırbistan ve Kazakistan arasında değiş**tiği** mesele yok.* In addition, verbs as verbal nouns with the form 'V-GA$_H$' in Kazakh also have two forms in Turkish: verbal nouns formed with the **-{D}{I}k** suffix and verbal adverbs formed with the **-y{A}r{A}k** suffix, such as the phrase *Мұғалім оқыт**қан**ыңды ұмытып кетті.*' 'A teacher forgot what to teach' *Hoca oku**duğ**unu unuttu* or *Hoca oku**yarak** unuttu*. Another example is a verbal noun in which the verb form in Kazakh with **–U** has two forms in Turkish as a gerund with the form **–m{A}** and a gerund infinitive with the form **-m{A}k**, such as the sentence *Полицияның жұмысы жаман адамдарды тұтқында**у***. 'The police's job is to arrest the bad men' which can be translated into Turkish as *Polisin işi kötü adamları tutukla**mak*** or *Polisin işi kötü adamları tutukla**ma***. We must write two rules for detecting these patterns. All the

circumstances mentioned are considered ambiguous texts from the source language (Kazakh) (see Fig. 1).

The dilemma concerning these ambiguous rules can be addressed by computing scores for sentence variants with a probabilistic language model and detecting the presumed rule by relying on the policy of maximum entropy. The probability of a target t being the translation for a word s in an SL context c is $p_s(t \mid c)$ (refer to Eq. (1)). Through the training process, every feature is assigned a weight λ^s and by using the maximum entropy classifier, the weights of ambiguous rules can be learned for each SL word. This classifier is then integrated into the translation model. The following equation summarizes how the most probable rule is selected:

$$p_s(t \mid c) = \frac{1}{Z} \exp \sum_{k=1}^{nF} \lambda_{k=1}^{s} h_k^s(t \mid c). \tag{1}$$

In Eq. (1), Z is a normalising constant of scores. Hence, the most possible translation can be encountered using Eq. (2).

$$\arg\max_{r \in T_{(s)}} p_s(t \mid c) = \arg\max_{r \in T_{(s)}} \sum_{k=1}^{nF} \lambda_{k=1}^{s} h_k^s(t \mid c) \tag{2}$$

In principle, the upper limit for the accomplishment of the system was TLM, because after obtaining all possible translations, we used TL information to decide which translations were better, and considered the main difference between TLM and unweighted systems. The process of applying rules is conducted in the following steps: First, rules are applied to input sentences, and rules matched to the input sentences are detected as the active features. Next, the probability $p_s(t|c)$ is computed for all active features (rules). We sum up the weights of the live rules for each TL translation of each SL word in place of selecting the longest rule using Eq. (1). The comprehensive architecture of the direct maximum entropy models is outlined in Fig. 2.

When the initial state q_0 is the only living state in the transducer, we retract and pick the translation with the highest total of weights as declared in Fig. 1. It should be noted that the main reason for using maximum entropy despite scoring our sentences with TLM, is that maximum entropy indicates which features are relevant and how to weight them. This means that we do not have to carry out all translations, but if we only use the language model, we should accomplish all the translations before scoring them. This saves time and increases performance.

3.2 Translation using Beam Search

We use a fundamental version of the beam search algorithm to find a translation that maximizes the conditional probability accorded by the maximum entropy model. First, we apply a set of ambiguous rules to some words, and then we obtain the weights of these words for every rule from models of the maximum entropy. Thereafter, we build a tree for these new words. The tree is based on vectors of rule indices along with the sum of their weights. Let us assume, that at any iteration, we have a set of rules applied to the same words. The beam tree would expand with the number of rules applied to the words. For example, if in any iteration a set of rules exists (r = 5 rules and w = 3 words) in

which every word matches three rules, three different translations are obtained for each word. In this case, the beam tree is expanded to have nine translations, and these trees merge with the existing beam tree. Then, we sort those nine translations by descending sum of weights, and reduce the beam tree to have no more than the beam size translations. Supposing that the beam size equals four, we remove the last five translations from the tree, and continue until we finish all the ambiguous rules, and the output is a tree with no more than the beam size translations. Finally, we obtain and output only the best translation.

3.3 Overview of Apertium

A typical translator built using the Apertium platform, including the translator described here, consists of a Unix-style pipeline or assembly line with the following modules (see Fig. 3).

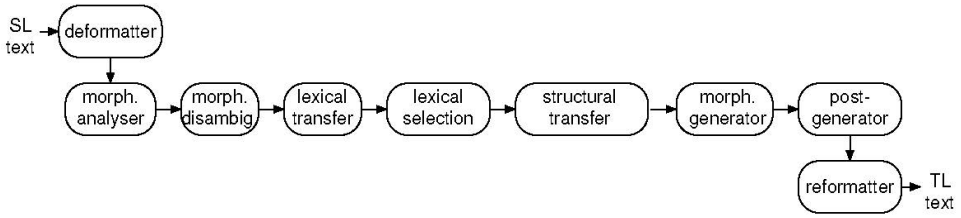


Fig. 3. The pipeline architecture of a typical Apertium MT system.

- **De-formatter:** Separates the text to be translated from the formatting tags.
- **Morphological analyzer:** Segments the source-language (SL) text in surface forms (SF) (words, or, where detected, multiword lexical units) and for each, delivers one or more lexical forms (LF) consisting of lemma (dictionary or citation form), lexical category (or part-of-speech) and inflection information.
- **Morphological disambiguator:** A morphological disambiguator that chooses the most adequate sequence of morphological analyses for an ambiguous sentence.
- **Lexical transfer:** This module reads each SL LF and delivers the corresponding target-language (TL) LF by looking it up in a bilingual dictionary encoded as an finite-state transducer compiled from the corresponding XML file. The lexical transfer module may return more than one TL LF for a single SL LF.
- **Lexical selection:** A **lexical selection** module selects the most adequate translation of ambiguous SL LFs based on context rules.
- **Structural transfer.** The structural transfer module applies a sequence of one or more finite-state constraint rules on the output of the lexical selection module so that it can select the left-right longest matching translation.
- **Morphological generator:** It transforms the sequence of TL LFs produced by the structural transfer to a corresponding sequence of TL SFs.
- **Post-generator:** Performs orthographic operations, for example elision (such as *da* + *il* = *dal* in Italian). This module has not been employed in our translator so far.
- **Reformatter:** De-encapsulates any format information.

Modules of the described pipeline are independent from each other and thus can rely on various programs and formalisms, and be of rule-based, statistical, or hybrid nature.

3.4 Corpora

To create the training corpora, we provide an SL corpus for training and a TL corpus for scoring. We used two freely available corpora: a dump of articles from Kazakh Wikipedia[2] with size 320 MB, 643.4 MB, and a single dump of articles from Turkish Wikipedia with size 440.6 MB were used for scoring TL corpus.

The training phase consists of setting up a co-occurrence model of SL lemmas (with the equivalent scores) for each translation sense managed by the MT system. As our application takes a sentence as input, we must break the corpus into sentences. To perform this task, we applied a rule-based sentence boundary detection tool called a pragmatic segmenter.[3]

Then, we randomly selected 1,000 sentences pairs for testing the performance of the system. Table 1 presents the statistics of the test corpora, specifically the number of sentences used for testing, and the number of tokens in the source and target sentences. The number of ambiguous tokens indicates the number of singular tokens with more than one likely translation. Then, we calculated the mean number of translations for each ambiguous word by dividing the number of ambiguous tokens by the entire number of tokens.

As we have compared our system with neural machine translation (openNMT) [12] system, and with another statistical machine translation system Moses [13], we used a parallel corpus for Kazakh and Turkish with size of 3.2 MB which is available online through OPUS[4].

Table 1 presents statistics for the parallel corpus that we used in NMT system, SMT system, and weighted system training. After splitting the corpus into training and development sets, we selected 5,000 sentence pairs for development (dev) and left the rest for training. Moreover, we calculated the number of ambiguous over the entire corpus. The number of ambiguous words indicates more than one possible translation with the number of particular tokens.

Table 1. Statistics of the test corpora and the training parallel corpora used by three systems (Weighted, NMT, and Moses).

	Lines	SL	TL	DEV	No.amb	%am-big
Train	62,893	266,555	285,648	5000	9.999	–
Test	1000	9,158	9,249	–	1,619	5.65

3.5 Reference Systems

We measured the performance of our method by comparing it to the consecutive reference (or baseline) systems:

- **Linguist-chosen defaults (unweighted).** A structural transfer in an Apertium language pair includes rules with the name of similar patterns. In this case, more than one rule can apply to the same token, which means that one sentence can have more than one

[2] https://kk.wikipedia.org
[3] https://github.com/diasks2/pragmatic_segmenter/tree/kazakh
[4] http://opus.nlpl.eu/

translation. If many possible translations of a structural form exist, then one must be signified as the default translation.

- **Random.** A set of transfer rules in an Apertium language pair contains rules with identical patterns. As more than one rule can apply to the same token. In case of many possible translations of a structural form, one must be selected randomly as a best translation.
- **Target language model (TLM).** A method of structural transfer that applies the extant MT system to produce all possible translations for a source sentence, and then scores these translations online on a model of the TL. The top scoring sentence is then output.

4. EVALUATION

The system was evaluated through translation quality measurement method, which calculates the error rate of the text produced by the system compared to the postedited version of the same system as a reference. A comparison of the evaluated system with reference systems shows that the error rate was quite low for a weighted system.

Furthermore, the system was evaluated and compared with state-of-the-art machine translation systems, for which we used our previous test data as presented in Table 1. We also compared the output of all systems with the postedited version of our weighted system.

We did not postedit each system independently because when we checked the output of other baseline systems, such as the Moses system, we found that most of the words were not in the vocabulary as presented in Table 2. Furthermore, over 50% were unknown words. Some sentences were not fully generated as target sentences.

Table 2. WER and PER; OOV is the number of out-of-vocabulary (unknown) words.

System	OOV%	WER (%)	PER (%)	BLEU (%)
Weighted	0.28	41.78	40.13	31.20
NMT	0.02	96.85	92.26	0.05
Moses	0.14	91.04	85.87	1.33

* WER and PER scores the reference system on the test corpora. The BLEU scores are computed against a *postedittted* reference translation.

Regarding NMT, the output (target) sentences were not related to the input sentences (source) sentences, which means that many of the sentences were translated arbitrarily and out of context. Additionally, the output sentences were not fully generated. Owing to these errors, postediting the output of each system independently is not a good decision, and would be detrimental to system performance.

4.1 Translation Quality

Translation quality was calculated using two different metrics: word error rate (WER) and position-independent word error rate (PER). These two metrics depend on the Levenshtein distance [14]. Metrics dependent on WER were selected to compare the system with systems based on comparable technology and to evaluate the usefulness of the system in a real-world setting, that is, to translate for distribution. Besides calculating

WER and PER for the weighted Kaz-ur MT system, we did the same for publicly available unweighted Kaz-Tur MT, TLM Kaz-Tur MT and Random Kaz-Tur MT systems. The policy was the same for all four systems. We picked a small number (9,158 tokens) of Kazakh text, which was a concatenation of several articles from Wikipedia, and translated it using the four MT systems. The output of each system was postedited independently to avoid bias in favour of one particular system. Next, we computed WER and PER for each using the each apertium-eval-translator tool[5], and we applied the widely used BLEU metric tool[6], which admirably tested the success rate of the system apropos of an approximate measurement of the final translation quality in a real system [15]. Note that the BLEU score is typically calculated by comparing the translation quality against a pretranslated reference translation. We have also used apertium-eval-translator for calculating WER and PER for all three systems (weighted, NMT, and SMT) as presented in Table 2.

4.2 Confidence Intervals

Confidence intervals for both metrics are calculated through bootstrap resampling as described by Koehn [16]. In all cases, bootstrap resampling is conducted for 1,000 repetitions. Wherever the $p < 0.05$ confidence intervals overlap, we achieve paired bootstrap resampling [16].

WER and PER scores with 75% confidence intervals for the reference systems on the test corpora. The BLEU scores are calculated against a *postedited* reference translation.

5. RESULTS AND DISCUSSION

When working with binary features, we used the execution of generalized iterative scaling available in the YASMET tool[7] to calculate feature weights.

Evaluation results are presented in Table 3, which compares the outcomes of the new approach (weighted) with the default (unweighted), randomly selected, and results are obtained by using the TL model online, for the language pair in Apertium with our two evaluation metrics. In addition, no large difference exists in the evaluation results because we selected the test data randomly from corpora of different articles, and not all sentences have ambiguous words. Furthermore, in some ambiguous words, the unweighted achieved a performance equal to that of the weighted. Significant enhancement with respect to TL model performance is expected as a result of the effective application that the maximum entropy model makes of information regarding appropriate SL contexts and their translations, through the weighting of features that represent those SL contexts over the entire corpus.

Table 3. WER and PER; OOV is the number of out-of-vocabulary (unknown) words.

System	OOV%	WER (%)	PER (%)	BLEU (%)
Weighted	0.28	24.72	24.51	55.28
Unweighted	0.28	32.14	31.88	42.28
TLM	0.28	28.85	28.48	48.31
Random	0.28	30.91	30.00	44.73

[5] http://wiki.apertium.org/wiki/apertiumevaltranslator
[6] https://www.letsmt.eu/Bleu.aspx
[7] https://wwwi6.informatik.rwthaachen.de/web/Software/YASMET.html

In addition, to evaluate our system on the parallel corpus and compare it with the performance of state-of-the-art MT systems trained on the same corpora, we trained the NMT and SMT baseline systems as the weighted system by taking the parallel dataset from the KDE4 corpus[8]. Table 2 exhibits the performance of the weighted system.

First, we compared the results of our weighted system with NMT[9], an NMT-small model from OpenNMT, with a framework employing neural translation. We trained the model at word level by using byte-pair encoding Second, we compared the weighted system with other publicly available SMTs such as the Moses system[10]. We used a phrase-based decoder in the Moses system [13], which allows us to create phrase-based systems using standard features that are usually used in current systems. The phrase-based decoder is used to train translation models for our language pair. Additionally, we trained 3-gram language models with Kneser-Ney smoothing using KenLM [17].

As shown in Table 2, the performance of the weighted system is much better than that of other baseline systems; the weighted system established a baseline of WER **41.78**, PER **40.13**, and BLEU **31.20**. For the out-of-vocabulary (unknown words) coverage in the corpus that we used for our experiments, the weighted system outperformed the NMT and Moses systems in WER, PER, and BLEU. One reason for the results is that the orthographic and dialectal variety of the texts used in the aligned corpus, may have prevented learning and generalization in the SMT and NMT systems. The weighted (RBMT) system is able to overcome this issue to some degree. Adding variants of frequent words is a simple issue, and one that we frequently addressed while developing the weighted system on the Wikipedia and news corpora.

The evaluation results of NMT were insufficient compared with our weighted system. Table 2 shows a very low BLEU score of 0.05, very high WER score of 96.85, and PER score of 92.26, which were obtained through our experiments in the NMT. However, most errors for the NMT system can be a factor in this event. Some sentences were much longer than the average appropriate length for NMT, thereby resulting in poor translation because encoder-decoder NMT models were unable to translate long sentences.

The NMT system performed poorly on lengthy sentences, but is relatively good up to a sentence length of approximately 60 words. As the NMT system produces short translations (length ratio 0.859, opposed to 1.024), the quality of these translations is drastically low [18]. Moreover, lack of data is a main reason for the poor quality of the NMT system. The figures obtained, given approximately 265,000 tokens of training data on each side seem to be consistent with experiments conducted on the relation of NMT performance and the amount of data [18]. Another reason for the poor performance was the relative lack of language standardization. Furthermore, the NMT system exhibited worse translation quality out of domain[11] than normal, which is a familiar challenge in translation in a different domain. Input words have various translations and their meanings are predicated in different styles. NMT is adapted for the sake of fluency. Although the output of the NMT system is sufficiently fluent, it is still completely unrelated to the input.

Most of the errors in the weighted system are due either to mistakes and gaps in the morphophonology components and disambiguation errors or input words being out of vocabulary. Furthermore, lexical selection was one of the causes of errors. The reason

[8] http://opus.nlpl.eu/
[9] http://opennmt.net/OpenNMTpy/
[10] http://www.statmt.org/moses/index.php

for this was that we made our system to select the first translation of an input word when more than one translation of the input word existed, and the first translation was not always suitable. The test corpus used for evaluation was not used while developing the RBMT system, including the training and development sets.

In case of SMT, we achieved a BLEU score of 1.33, WER score of 91.04, and PER score of 85.87 respectively. These results are lower than those of our weighted system. The error causes include the following reasons:

The main error category is a factor of the scarcity of data used during our experiments. Furthermore, the performance declines with a limited amount of parallel data. Big data is expected to yield better performance. Another reason was the existence of low-frequency words and word formation errors, which characterize the morphological richness of Kazakh and Turkish, and that negatively affect the quality of the SMT system; this system performs poorly on morphologically rich languages [22]. In addition, we found that the translation was less fluent. Some errors were related to accuracy, particularly mistranslation and omission.

6. CONCLUDING REMARKS

When a sufficiently large, freely available parallel corpus exists, rule-based MT is not competitive with corpus-based approaches, such as NMT and SMT. However, when only a small parallel corpora exists, the method can be competitive, especially between closely related languages or languages with non-trivial morphology, such as Kazakh and Turkish.

In this paper, we presented a method shown to be better than state-of-the-art RBMT for Kazakh and Turkish. This method also improves the current structural transfer in RBMT and can be trained in an unsupervised manner, that is, without using an annotated corpus (in this circumstance, a word-aligned bilingual corpus); one will only need an SL corpus, a statistical TL model, and the RBMT system itself. The method accepts input as the part-of-speech tagged source text in which each word is annotated with the translations provided by the bilingual dictionary in the system, thereby making the method suitable to nearly any RBMT system. The system uses maximum-entropy formalism for structural transfer. Instead of counting actual transfer rule selection events in an annotated corpus, the system counts fractional occurrences of these events as supposed by a TL model. The method is evaluated extrinsically, by measuring the quality of MT.

We evaluated our system using part of the monolingual data dump of articles from Kazakh Wikipedia, and compared the results with that of NMT and SMT systems trained on that corpus. The results indicate that even in 2019, RBMT can be used between closely related, morphologically rich languages when resources are insufficient to train the cutting edge in NMT and SMT. These evaluation methods have shown that the performance of the weighted system was better than of the other systems. Results of the Kaz-Tur pair using the Apertium MT system show that the method obtains results similar to or better than those obtained at greater cost by scoring an exponential number of transfer rule selections for each sentence using the TL model online.

We aim to continue development of the weighted transfer module to apply only chunker transfer rules (patterns of words), which will be conducted by extending the

[11] In NMT, a domain can be described by a corpus from a specific source, and may diverge from other domains in topic, genre, style, level of formality, and other factors.

module into other stages of structural transfer such as interchunk and postchunk transfer rules (patterns of chunks). Both interchunk and postchunk transfer rules are analogous to the chunker, but with certain dissimilarities. The module has already been integrated into Apertium, and is ready for use as free/open-source software under the GNU GPL. The entire system can be downloaded from GitHub.

ACKNOWLEDGEMENTS

The present work was supported by the Apertium organization. We are thankful to the community, who provided expertise that greatly assisted the research.

REFERENCES

1. M. R. Costa-Jussà and M. Farrús, "Statistical machine translation enhancements through linguistic levels: A survey," *ACM Computing Surveys*, Vol. 46, 2014, p. 42.
2. F. J. Och, "Statistical machine translation: Foundations and recent advances," *Tutorial at MT Summit*, 2005.
3. N. Kalchbrenner and P. Blunsom, "Recurrent continuous translation models," in *Proceedings of Conference on Empirical Methods in Natural Language Processing*, 2013, pp. 1700-1709.
4. I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in Neural Information Processing Systems*, 2014, pp. 3104-3112.
5. M. R. Costa-Jussa and J. A. Fonollosa, "Latest trends in hybrid machine translation and its applications," *Computer Speech & Language*, Vol. 32, 2015, pp. 3-10.
6. T. Ehara, "System combination of rbmt plus spe and preordering plus smt," in *Proceedings of the 2nd Workshop on Asian Translation*, 2015, pp. 29-34.
7. M. L. Forcada, M. Ginestí-Rosell, J. Nordfalk, J. O'Regan, S. Ortiz-Rojas, P.-O. Juan, Antonio, F. Sánchez-Martínez, G. Ramírez-Sánchez, and F. M. Tyers, "Apertium: a free/open-source platform for rule-based machine translation," *Machine Translation*, Vol. 25, 2011, pp. 127-144.
8. S. Bayatli, S. Kurnaz, I. Salimzianov, J. N. Washington, and F. M. Tyers, "Rule-based machine translation from kazakh to turkish," in *European Association for Machine Translation*, 2018, pp. 49-58.
9. I. Salimzyanov, J. N. Washington, and F. M. Tyers, "A free/open-source Kazakh-Tatar machine translation system," in *Machine Translation Summit XIV*, 2013, pp. 175-182.
10. A. Sundetova, M. L. Forcada, and F. M. Tyers, "A free/open-source machine translation system for English to Kazakh," in *Proceedings of the 3rd International Conference on Turkic Languages Processing*, 2015, pp. 78-91.
11. N. Habash, D. Bonnie, and M. Christof, "Symbolic-to-statistical hybridization: extending generation-heavy machine translation," *Machine Translation*, Vol. 23, 2009, pp. 23-63.
12. G. Klein, Y. Kim, Y. Deng, J. Senellart, and A. M. Rush, "Opennmt: Open-source

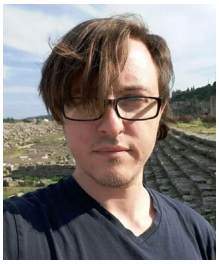
toolkit for neural machine translation," in *Proceedings of the 55th ACL System Demonstrations*, 2017, pp. 67-72.
13. P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens *et al.*, "Moses: Open source toolkit for statistical machine translation," in *Proceedings of the 45th ACL on Interactive Poster and Demonstration Sessions*, 2007, pp. 177-180.
14. V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," in *Soviet Physics Doklady*, 1966, pp. 707-710.
15. K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, 2002, pp. 311-318.
16. P. Koehn, "Statistical significance tests for machine translation evaluation," in *Proceedings of Conference on Empirical Methods in Natural Language Processing*, 2004, pp. 388-395.
17. K. Heafield, "Kenlm: Faster and smaller language model queries," in *Proceedings of the 6th Workshop on Statistical Machine Translation*, 2011, pp. 187-197.
18. P. Koehn and R. Knowles, "Six challenges for neural machine translation," in *Proceedings of the 1st Workshop on Neural Machine Translation*, 2017, pp. 28-39.

Sevilay Bayatli received her master degree in Computer Science from Gazi university. She is a Ph.D. student at Altınbaş University. Her research interests are related to the linguistics of Turkic languages.

Sefer Kurnaz is an Assistant Professor at Altınbaş University. He has mainly specialized in space-technology applications. He graduated with a Ph.D. from İstanbul University, specializing in computer science.

Aboelhamd Ali studied Computers and Systems Engineering at Faculty of Engineering, Alexandria University. He is a Software Engineer at Brightskies Technologies.

Jonathan N. Washington is an Assistant Professor of Linguistics at Swarthmore College. His research focuses on Turkic languages, with emphasis on phonetics, phonology, historical linguistics, and language technology.



Francis M. Tyers is an Assistant Professor at Indiana University and Adjunct Professor at Higher School of Economics. His interests are in language technology for indigenous and marginalized languages.