Rule-based machine translation from Kazakh to Turkish

Sevilay Bayatli
Dept. Elec. and Computer Engineering
Altınbaş Üniversitesi
sevilaybayatli@gmail.com

Sefer Kurnaz
Dept. Elec. and Computer Engineering
Altınbaş Üniversitesi
sefer.kurnaz@altinbas.edu.tr

Ilnar Salimzianov
School of Science and Technology
Nazarbayev University
ilnar@selimcan.org

Jonathan North Washington
Linguistics Deptartment
Swarthmore College
jonathan.washington@swarthmore.edu

Francis M. Tyers
School of Linguistics
Higher School of Economics
ftyers@hse.ru

Abstract

This paper presents a shallow-transfer machine translation (MT) system for translating from Kazakh to Turkish. Background on the differences between the languages is presented, followed by how the system was designed to handle some of these differences. The system is based on the Apertium free/open-source machine translation platform. The structure of the system and how it works is described, along with an evaluation against two competing systems. Linguistic components were developed, including a Kazakh-Turkish bilingual dictionary, Constraint Grammar disambiguation rules, lexical selection rules, and structural transfer rules. With many known issues yet to be addressed, our RBMT system has reached performance comparable to publicly-available corpus-based MT systems between the languages.

1 Introduction

In this paper we present a prototype shallow-transfer rule-based machine translation system using the Apertium free/open-source machine translation platform (Forcada et al., 2011) for translating from Kazakh to Turkish.

One of the most common criticisms towards Rule-Based Machine Translation (RBMT) regards the amount of work necessary to build a system for a new language pair (Arnold, 2003). In fact, in a traditional scenario, linguists with expertise in the source and target language need to manually build all the dictionary entries and transfer rules. Conversely, in a corpus-based/statistical MT approach (Koehn, 2010), no such effort is required as the system can be automatically built from parallel corpora providing they exist. If parallel corpora do not exist, then we see two options that remain. The first is to create a new parallel corpus, either by translating millions of words from scratch (requiring effort from translators),[1] or by finding parallel text online and processing it (requiring effort from programmers). The second option is to build a rule-based machine translation system (requiring effort from linguists). The most labour-intensive of these approaches is to translate the data from scratch, although this might be practical in certain situations. Building a rule-based machine translation system and finding and processing parallel texts from the internet are, given equal available expertise, around equally time consuming. As large, freely available parallel corpora are not known to exist between Kazakh and Turkish and we were interested in structural differences between these two languages — and producing new linguistic resources, we chose to build a rule-based MT system. Kazakh and Turkish are different enough that native speakers are not able to make sense of the other language, but also share similar enough structure that an RBMT system is feasible with some level of linguistic knowledge.

This paper demonstrates that, with many known issues yet to be addressed, our RBMT system has already reached performance comparable to publically available SMT systems between the languages. This has been accomplished solely with open source tools and some level of linguistic knowledge about

[1] The millions of words number is taken from Koehn and Knowles (2017) who compare neural MT against phrase-based SMT for English–Spanish for 0.4 million to 385.7 million words. Even for these morphologically-poor languages, even with one million words the performance is poor.

Pérez-Ortiz, Sánchez-Martínez, Esplà-Gomis, Popović, Rico, Martins, Van den Bogaert, Forcada (eds.)
Proceedings of the 21st Annual Conference of the European Association for Machine Translation, p. 49–58
Alacant, Spain, May 2018.

the two languages, and without large parallel corpora or machine learning algorithms (although some of the components in our opinion could be significantly improved by using them; see Section 6 for more details).

The paper will be laid out as follows: Section 2 gives a short review of some previous work in the area of Turkic-Turkic language machine translation and an overview of other publically available Kazakh-Turkish machine translators; Section 3 introduces Kazakh and Turkish and compares their grammar; Section 4 describes the system and the tools used to construct it; Section 5 gives a preliminary evaluation of the system; Section 6 describes our aims for future work; and finally Section 7 contains some concluding remarks.

2 Previous work

Within the Apertium project, there is ongoing work on building MT systems (and thus underlying components such as morphological transducers) for translating between two Turkic languages, between a Turkic language and Russian or between a Turkic language and English. Among released MT systems there are: Kazakh-Tatar (Salimzyanov et al., 2013), Tatar-Bashkir (Tyers et al., 2012b), Crimean Tatar-Turkish and English-Kazakh (Sundetova et al., 2015) MT systems.

Several other MT systems have been reported that translate between Turkish and other Turkic languages, including Turkish–Crimean Tatar (Altıntaş, 2001), Turkish–Azerbaijani (Hamzaoglu, 1993), Turkish–Tatar (Gilmullin, 2008), and Turkish–Turkmen (Tantuğ et al., 2007a,b) MT systems. As for the systems for translating to/from Kazakh (besides the ones already mentioned above), there is a bidirectional Kazakh-English machine translation system (Tukeyev et al., 2011) which uses a link grammar and statistical approach. None of these MT systems to our knowledge have been released to a public audience.

Altenbek and Xiao-long (2010) propose a segmentation system for inflectional affixes of Kazakh. Makhambetov et al. (2015), Kessikbayeva and Cicekli (2014) and Kairakbay and Zaurbekov (2013) present work on Kazakh morphological analysis.

Both Kazakh and Turkish are among the languages supported by Google Translate[2] and Yandex Translate[3] tools.

3 The languages

Kazakh is classified as a member of the Northwestern (or Kypchak) branch of the Turkic language family. It is primarily spoken in Kazakhstan, where it is the national language, sharing official status with Russian. Large communities of native speakers also exist in China, neighbouring Central Eurasian republics, and Mongolia. The total number of speakers is at least 10 million people (Simons and Fennig, 2018). The present-day Kazakh Cyrillic alphabet consists of 42 letters, 33 of which are letters found in the Russian alphabet. There are controversial plans to transition to a Latin alphabet by 2025.

Turkish is classified as a member of the Southwestern (or Oghuz) branch of Turkic language family. With over 70 million L1 speakers (Simons and Fennig, 2018), it is the Turkic language spoken by the most people. The Turkish Latin alphabet contains 29 letters.

Kazakh and Turkish exhibit agglutinative morphology, meaning that word forms may consist of a root and a series of affixes.

An MT system between Kazakh and Turkish is potentially of great use to the language communities. Automatic MT can save time and money over going through e.g. Russian or English (and the system is much easier to develop).

We continue with a brief overview of some differences in phonology, orthography, morphology and syntax of the languages. A complete and detailed comparison is out of scope of this work.

3.1 Phonology and orthography

Differences in phonology and orthography are less relevant for this work because relatively high-coverage morphological transducers were available for both of the languages when we started working on the translator. The mutual intelligibility of Kazakh and Turkish, both in their spoken and written forms, is rather low, despite much similar morphology and the existence of many cognates, often with similar meanings.

3.2 Morphology and syntax

3.2.1 Verbals

There are verbal tenses and moods common to both Kazakh and Turkish, like the definite past tense, the imperative mood, and the conditional mood. There are also quite a few differences. For example, Kazakh lacks the definite future tense affix **-{y}{A}c{A}k** known in Turkish; but has the so

[2] `http://translate.google.com`
[3] `http://translate.yandex.com`

called *goal oriented future* tense, absent in Turkish: e.g., the Kazakh verb form *бармақпын* ‘I intend to go’ can be translated into Turkish as *gitmeyi düşünüyorum* ‘I intend to go’. Another example of an affix found in one language but not in the other is the affix **-{D}{A}й** in Kazakh, which follows nouns and numbers and indicates resemblance, and can often be translated as the postposition *gibi* ‘like’ in Turkish.

Kazakh has several auxiliary verbs which are used for constructing analytic verbal forms. Four of them, the auxiliary verbs *жатыр, отыр, жүр, тұр* are used to construct the present continuous tense (Muhamedow, 2016), as in the collocation *жауып жатыр* ‘is raining’, translated to Turkish as *yağıyor* ‘is raining’. There are many other cases (i.e., not just due to analytic tenses) in which sequences of two or more Kazakh verbs map to a single verb in Turkish, as in the case of the expression *қуанып кетті* ‘gladdened’, which is translated as *neşelendi* ‘gladdened’ in Turkish.

In the case of non-finite forms, there are one-to-many correspondences. For instance, Kazakh past verbal adjectives (participles) formed with the **-{G}{A}н** suffix can be translated into Turkish in at least three ways: as past verbal adjective with the **-m{I}ş** suffix, as a subject-relative verbal adjective formed with the **-{y}{A}n** suffix or as a past verbal adjective formed with the **-{D}{I}k** suffix. As an example, the Kazakh sentence *Сербия мен Қазақстан арасында шешілме**ген** мәселе жоқ.* ‘There aren’t any **unresolved** issues between Serbia and Kazakhstan’ can be translated into Turkish as *Sırbistan ve Kazakistan arasında çözümlenme**miş** mesele yok.*, whereas the sentence *Екі мемлекет басшылары шағын және кеңейтіл**ген** құрамда келіссөздер жүргізді.* ‘The two leaders held talks in small and **expanded** format.’ in the parallel corpus we constructed (see Section 4.3) is translated as *İki memleket başkanları küçük ve genişletil**diği** kapsamda müzekereler yönetti.*

Similarly, the Kazakh imperfect verbal adjective formed with the suffix **-{E}т{I}н** is translated as either a subject-relative verbal adjective formed with the **-{y}{A}n** suffix or as future verbal adjective constituted with **-{y}{A}c{A}k** suffix. For example, the Kazakh phrase *сөйлейтін* can be translated as *konuşacak* ‘which will speak’ or as *konuşan* ‘(which is) speaking’.

3.2.2 Nominals

Another example of a morphological difference between Kazakh and Turkish is the presence of a four-way distinction in Kazakh’s 2nd person system (both pronouns and agreement suffixes). In other words, in Kazakh there is a distinct word for all combinations of [±plural, ±formal] (Muhamedow, 2016), whereas the Turkish 2nd person singular formal pronoun coincides with the 2nd person plural informal and 2nd person plural formal pronouns, as summarized in Table 1 (both *siz* and *sizler* are used as the plural formal pronoun in Turkish).

	Kazakh		**Turkish**	
	-PLUR	+PLUR	-PLUR	+PLUR
-FRM	сен	сендер	sen	siz
+FRM	сіз	сіздер	siz	siz/sizler

Table 1: Second person personal pronouns in Kazakh and Turkish. Note the extra distinctions in the Kazakh forms.

All of the differences or one-to-many/many-to-one correspondences in morphology and syntax described in this and the preceding subsections are relevant for a shallow-transfer RBMT because to handle them is the main job of the transfer component.

4 System

Our machine translation system is based on the Apertium MT platform (Forcada et al., 2011).[4] The platform was originally aimed at the Romance languages of the Iberian peninsula, but has also been adapted for other, more distantly related, language pairs. The whole platform, both programs and data, are licensed under the Free Software Foundation’s General Public Licence[5] (GPL) and all the software and data for the completed supported language pairs (and the other pairs being worked on) is available for download from the project website.

4.1 Architecture of the system

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. 1. In Table 2 you can see an example of how a Kazakh sentence passes through the pipeline):

- **De-formatter.** Separates the text to be translated from the formatting tags. Formatting tags

[4] http://www.apertium.org

[5] http://www.fsf.org/licensing/licenses/gpl.html

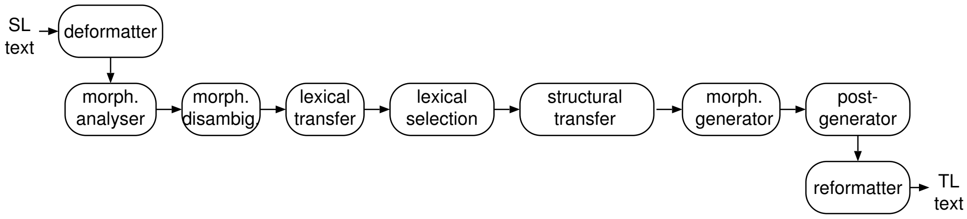

Figure 1: The pipeline architecture of a typical Apertium MT system.

are encapsulated in brackets so they are treated as "superblanks" that are placed between words in such a way that the remaining modules see them as regular blanks.

- **Morphological analyser.** 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, in case of the Kazakh-Turkish translator based on the Constraint Grammar (CG) formalism (Karlsson et al., 1995), 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 Tyers et al. (2012a) chooses, based on context rules, the most adequate translation of ambiguous SL LFs.

- **Structural transfer.** A structural transfer module, which performs local syntactic operations, is compiled from XML files containing rules that associate an action to each defined LF pattern. Patterns are applied left-to-right, and the longest matching pattern is always selected.

- **Morphological generator.** It transforms the sequence of target–language LFs, produced by the structural transfer, to a corresponding sequence of target–language 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.

The Apertium platform provides what can be called a 'vanilla' program and a formalism for describing linguistic data (if the module in question requires it) for each of the modules. We want to emphasise though that modules of the pipeline just described are independent from each other and thus can rely on different programs, different formalisms, and be of rule-based, statistical or hybrid nature. For example, Constraint Grammar-based morphological disambiguator can be considered a drop-in replacement for the Hidden Markov Model-based statistical tagger found in a few other Apertium MT systems. So are the formalisms used for morphological transducers which are described next.

4.2 Morphological transducers

The morphological transducers are based on the Helsinki Finite State Toolkit (Linden et al., 2011) – a free/open-source reimplementation of the Xerox finite-state tool chain, popular in the field of morphological analysis. It implements both the **lexc** formalism for defining lexicons, and the **twol** and **xfst** formalisms for modeling morphophonological rules. This toolkit has been chosen as it — or the equivalent XFST — has been widely used for other Turkic languages (Cöltekin 2010; Altintas and Cicekli 2001; Washington et al. 2012; Tantuğ et al. 2006; Tyers et al. 2012b,Washington et al. 2014, Çöltekin 2014) and is available under a free/open-source licence. The morphologies of both languages are implemented in lexc, and the morphophonologies of both languages are implemented in twol. The same lexc and twol files are used to compile both the morphological analyser and the morphological generator for each language.

(Kazakh) Input	Біз жаттығулар барысын мұқият бақылап отырдық.
Mor. analysis	^Біз/біз<prn><pers><p1><pl><nom>$ ^жаттығулар/жаттығу<n><pl><nom>/жаттығу<n><pl><nom>+e<cop><aor><p3><sp>$ ^барысын/бары<n><px3sp><acc>/барыс<n><px3sp><acc>$ ^мұқият/мұқият<adj>/мұқият<adv>/мұқият<adj>+e<cop><aor><p3><sp>$ ^бақылап/бақыла<v><tv><prc_perf>/бақыла<v><tv><gna_perf>$ ^отырдық/отыр<vaux><ifi><p1><pl>/отыр<v><iv><ifi><p1><pl>$^./.<sent>$
Mor. disambig	^Біз<prn><pers><p1><pl><nom>$ ^жаттығу<n><pl><nom>$ ^бары<n><px3sp><acc>$ ^мұқият<adv>$ ^бақыла<v><tv><prc_perf>$ ^отыр<vaux><ifi><p1><pl>$ ^.<sent>$
Lex. transfer	^Біз<prn><pers><p1><pl><nom>/Biz<prn><pers><p1><pl><nom>$ ^жаттығу<n><pl><nom>/çalışma<n><pl><nom>/egzersiz<n><pl><nom>$ ^бары<n><px3sp><acc>/süreç<n><px3sp><acc>$ ^мұқият<adv>/dikkatlice<adv>$ ^бақыла<v><tv><prc_perf>/gözlemle<v><tv><prc_perf>$ ^отыр<vaux><ifi><p1><pl>/<ifi><p1><pl>/otur<v><iv><ifi><p1><pl>$^.<sent>/.<sent>$
Structural transfer	^Biz<prn><pers><p1><pl><nom>$ ^çalışma<n><pl><nom>$ ^süreç<n><px3sp><acc>$ ^dikkatlice<adv>$ ^gözlemle<v><tv><ifi><p1><pl>$^.<sent>$
Mor. generation	Biz çalışmalar sürecini dikkatlice gözlemledik.

Table 2: Translation process (from Kazakh to Turkish) for the phrase *Біз жаттығулар барысын мұқият бақылап отырдық*. 'We carefully followed the work process.' Some analyses are omitted for reasons of space. Note how a transfer rule has transformed a participle + auxiliary construction of Kazakh, *бақылап отырдық* 'we followed', to an analytic construction in Turkish, *gözlemledik* 'we followed'.

The Kazakh morphological transducer used in this work was presented in (Washington et al., 2014). Turkish morphological transducer also comes from the Apertium project. It has not been described in a published work yet. Both transducers were extended to support all stems from the bilingual lexicon we constructed.

We decided to use the Turkish morphological transducer developed in the Apertium project and not the also free/open-source TRMorph (Çöltekin, 2014), because the tagset used in the former is more consistent with morphological transducers developed in the Apertium project for other Turkic languages, including the Kazakh transducer. The consistency of the tagset allows to keep the transfer module relatively simple and pay more attention to the actual differences in the grammar of the languages rather than on differences in the tagset used.

4.3 Bilingual lexicon

The bilingual lexicon currently contains 7,385 stem-to-stem correspondences and was built mostly by hand in the following way. We assembled a parallel Kazakh-Turkish corpus. For this we took all sentences from the Kazakh treebank (Tyers and Washington, 2015) — approximately one thousand sentences — and translated them manually to Turkish. Then, these Kazakh and Turkish sentences were analysed with the `apertium-kaz` and `apertium-tur` morphological transducers. This provided the lemma and the part of speech tag for most of the surface forms in the corpora. The lemmas which were not already in the monolingual lexicons were added to them, and corresponding words were added to the bilingual lexicon. In addition, some of the stems present in the Kazakh lexc file but not found in the parallel corpus were translated into Turkish and added to the bilingual dictionary. Because of the similarity of the languages, the majority of entries in the bilingual dictionary (a file in an XML-based format) are one-to-one mappings of stems, but there are ambiguous translations. For example, the Kazakh word 'азамат' has two translations in Turkish: 'sivil' and 'vatandaş', as shown in Fig. 2.

4.4 Rules

The note made at the end of Section 4.1 on replacability of the components aside, Apertium is primarily a rule-based MT system. Not counting morphophonology (morphotactics) rules required by HFST-based morphological transducers, there are three main categories of rules in our system — morphological disambiguation rules, lexical selection rules and transfer rules. A description of each follows.

```
<e><p><l>қас<s n="n"/></l>       <r>ruh<s n="n"/></r></p></e>
<e><p><l>азамат<s n="n"/></l>    <r>sivil<s n="n"/></r></p></e>
<e><p><l>азамат<s n="n"/></l>    <r>vatandaş<s n="n"/></r></p></e>
<e><p><l>үлкен<s n="adj"/></l>  <r>büyük<s n="adj"/></r></p></e>
<e><p><l>ұлттық<s n="adj"/></l> <r>ulusal<s n="adj"/></r></p></e>
<e><p><l>дауа<s n="n"/></l>      <r>çare<s n="n"/></r></p></e>
<e><p><l>дауа<s n="n"/></l>      <r>ilaç<s n="n"/></r></p></e>
<e><p><l>дауа<s n="n"/></l>      <r>çözüm<s n="n"/></r></p></e>
<e><p><l>шешім<s n="n"/></l>     <r>çözüm<s n="n"/></r></p></e>
<e><p><l>шешім<s n="n"/></l>     <r>karar<s n="n"/></r></p></e>
```

Figure 2: Example entries from the bilingual lexicon. Kazakh is on the left, and Turkish on the right. Each stem is accompanied by a part-of-speech tag and there may be many–many correspondences between the stems.

4.4.1 Morphological disambiguation rules

The system has a morphological disambiguation module in the form of a Constraint Grammar (CG) (Karlsson et al., 1995). The version of the formalism used is vislcg3.[6] The goal of the CG rules is to select the correct morphological analysis when there are multiple analyses. We used the Kazakh CG previously developed partially by the authors of this paper and partially by other Apertium contributors. At the time of this writing the file contains 164 rules. Due to closeness of the languages, the majority of ambiguity may be passed through from one language to the other.

4.4.2 Lexical selection rules

In general, lexical selection rules are necessary to handle one-to-many correspondences of the bilingual lexicon. While many lexical items have a similar range of meaning, lexical selection is sometimes necessary when translating between Kazakh and Turkish as well. For example, the Kazakh word *ат* has two meanings: *ат* 'name' and *ат* 'horse' and can be translated into Turkish as either *ad* 'name' or *at* 'horse'. A lexical selection rule chooses the translation *at* 'horse' if the immediate context includes a word *ұста* 'hold'. Another example is the word *жарық*, which as a noun can mean either 'light' or 'crack'. It is translated to Turkish by default as *ışık* 'light', and is translated as *yarık* 'crack' only in the immediate context of words like *есік* 'door' and *қабырға* 'wall'. A relatively small number of 92 lexical selection rules were developed and added to the system. The lexical selection module we used (Tyers et al., 2012a) allows inferring such rules automatically from a parallel corpus, but we have not employed this feature of it yet.

4.4.3 Structural transfer rules

Apertium, as a rule, translates lemmas and morphemes one by one. Obviously, this does not always work, even for closely related languages. Structural transfer rules are responsible for modifying morphology or word order in order to produce "adequate" target language.

As seen in Table 2, the structural transfer module takes a sequence of (source language lexical form — target language lexical form) pairs in the following format: `ˆSL-lemma<SL-tag1><SL-tag2><…><SL-tagN>` `/TL-lemma<TL-tag1><TL-tag2><…><TL-tagN>$` TL lemma and tags are provided by the preceding two modules — lexical transfer and lexical selection. The lexical transfer module looks up the TL lemma and usually the first one or two tags (read: part of speech tag) in the bilingual transducer, the rest of the tags are carried over from the SL.

Figure 3 gives an example of a transfer rule. Any transfer rule consists of two core parts — of a pattern and an action. In this case, the pattern named "gpr_impf" matches Kazakh verbal adjectives formed with the **-{E}т{I}н** affix (for reasons of space, we omitted the definition of the pattern itself). Recall from Section 3.2.1 that Kazakh verbal adjectives ending in **-{E}т{I}н** have two possible translations in Turkish — either with a verbal adjective ending in **-{y}{A}n** suffix or with a verbal adjective ending in **-{A}c{A}k** suffix. The rule in Figure 3 replaces the `<gpr_impf>` tag on the TL side with the `<gpr_rsub>` tag, which corresponds to a **-{y}{A}n** verbal adjective in Turkish. In addition, transfer rules perform chunking, and later transfer stages can operate on chunks of words as if they were single words, but we will not discuss chunking-based rules here since this technique is currently not employed in the Kazakh-Turkish translator.

The patterns are matched on the SL side by the

[6] `http://beta.visl.sdu.dk/constraint_grammar.html`

```
<rule comment=”REGLA: gpr_impf-5” > <!--сөйлейтін -> konuşan -->
  <pattern><pattern-item n=”gpr_impf”/></pattern>
  <action>
    <let><clip pos=”1” side=”tl” part=”a_impf”/><lit-tag v=”gpr_rsub”/></let>
    <out>
      <chunk name=”gpr” case=”caseFirstWord”>
        <tags><tag><lit-tag v=”SV”/></tag></tags>
        <lu><clip pos=”1” side=”tl” part=”whole”/></lu>
      </chunk>
    </out>
  </action>
</rule>
<rule comment=”REGLA: ger_perf-7” > <!--білгендік -> bildik -->
 <pattern><pattern-item n=”ger_perf”/></pattern>
 <action>
   <let><clip pos=”1” side=”tl” part=”ger_prf”/><lit-tag v=”ger_past”/></let>
   <out>
    <chunk name=”v” case=”caseFirstWord”>
      <tags><tag><lit-tag v=”SV”/></tag></tags>
      <lu><clip pos=”1” side=”tl” part=”whole”/></lu>
    </chunk>
   </out>
 </action>
</rule>
```

Figure 3: Examples of a transfer rule. The first rule translates Kazakh -{E}т{I}н verbal adjectives with -(y)An verbal adjectives by replacing the `<gpr_impf>` tag on the TL side with the `<gpr_rsub>` tag. The second rule transfer Kazakh -{G}{A}нл{I}{K} verbal noun (gerund) into -{D}{I}k verbal noun (Gerund) by replacing the `<ger_perf>` tag on the TL side with the `<ger_past>` tag.

"left–to–right, longest–match" principle. For instance, Kazakh determiner–adjective–noun phrase *Бұл үлкен жетістік* 'This big success' will be matched and processed by a rule having determiner–adjective–noun sequence as its pattern, and not by e.g. a rule matching determiner–adjective or determiner–noun sequences. If there are ties, the rule which is placed higher in the source file is applied first.

Since Kazakh and Turkish have similar syntax, most of the rules are not about reordering but about altering the tags carried over from SL LFs to TL LFs. We used the parallel corpus described in Section 4.3 as a development set for writing and refining transfer rules. In all, we have defined 76 structural transfer rules.

5 Evaluation

The system has been evaluated in two ways. The first is its coverage. The second is translation quality — the error rate of two pieces of text produced by the system when comparing with postedited versions of them.

5.1 Coverage

Coverage of the system was calculated over three freely-available corpora — a dump of articles from Kazakh Wikipedia, a Kazakh translation of the Quran, and the Kazakh translation of the Bible. The corpus extracted from Kazakh Wikipedia[7] contains 22,515,314 words (1,404,467 sentences). Wikipedia is one of the major uses for Apertium translators, especially since some of these are used by the Wikimedia Content Translation Tool[8].

Corpus	Tokens	Coverage (%)
Wikipedia	22,515,314	83.42
Quran.altay	107,451	70.30
Bible.kkitap	577,070	80.89

Table 3: Coverage of the Kazakh-Turkish MT system

The coverage is the percentage of tokens in running text which are translated by the MT system. Apertium MT systems mark tokens that could not be analysed with a special sign, thus coverage was calculated by dividing the number of tokens without that special sign by the total number of words in the text. The coverage results are presented in Table 3. As seen in the table, the coverage of the system is not very high, with an average of 78.20% .

[7] https://kk.wikipedia.org/; kkwiki-20180320-pages-articles.xml.bz2

[8] https://www.mediawiki.org/wiki/Content_translation

5.2 Translation quality

The translation quality was measured using two metrics, the first was word error rate (WER), and the second was position-independent word error rate (PER). Both metrics are based on the Levenshtein distance (Levenshtein, 1965). Metrics based on word error rate were chosen as to be able to compare the system against systems based on similar technology, and to assess the usefulness of the system in a real setting, that is of translating for dissemination.

Besides calculating WER and PER for our Kazakh-Turkish MT system, we did the same for two other publically available Kazakh-Turkish MT systems — from Google Translate and Yandex Translate. The procedure was the same for all three. We took a small (1,025 tokens) Kazakh text, which was a concatenation of several articles from Wikipedia and translated it using the three MT systems. The output of each system was postedited independently to avoid biasing in favour of one particular system. Then we calculated WER and PER for each using the `apertium-eval-translator` tool[9] and BLEU using the `mteval-v13a.pl` script.[10] Note that BLEU score is typically calculated by comparing against a pre-translated reference translation, where here we calculate against posteditted reference translations for each of the systems.

5.2.1 Results

Table 4 shows the results obtained for all three systems — Google, Yandex and Apertium. Google's[11] MT system has the lowest WER. Apertium has a comparable WER despite having much higher number of OOV words. Yandex Translate's[12] WER is higher, but PER is similar to the other two.

These numbers can be compared with scores for other translators based on the Apertium platform. For example, the Kazakh–Tatar system described in (Salimzyanov et al., 2013) achieves post-edition WER of 15.19% and 36.57% over two texts of 2,457 words and 2,862 words respectively. The Tatar–Bashkir system in (Tyers et al., 2012b) reports WER of 8.97% over a small text of 311 words and WER of 7.72% over another text of 312 words. The higher word error rate can be explained by the fact that Kazakh and Turkish are more distantly related than Tatar and Kazakh or Bashkir.

5.3 Error analysis

The majority of remaining errors are mostly due to a lot of unknown words (because the relatively low number of words in the bilingual dictionary), and disambiguation errors.

6 Future work

We intend to continue the development of the system to improve the quality of the translations. There are a number of areas where we believe that more work would yield better results, among which are the following:

- **Coverage**. By expanding the dictionaries with new lists of stems, and providing bilingual correspondences, the error rate will decrease and, consequently, there will be less post-editing work necessary (and translations will be much more intelligible). The principle issue here is adequate Kazakh–Turkish lexicography which is an under-investigated area.

- **Ambiguous transfer rules**. The "left-to-right, longest match" principle by which structural transfer rules are applied (which implies that if there are several rules with the same pattern, only one of them will apply — the first one that matches that pattern), although makes it easy to change the behaviour of the system (simply by reordering rules), in our opinion, is quite limiting. In particular, it limits the ways how one-to-many correspondences can be handled (several examples of such cases were given in Section 3), essentially forcing the developer to hard code one of the several possible translations as the default one and checking the surrounding lemmas or other features in the rule manually if he wishes to select an alternative translation. We conceive a method of selecting the most adequate rules for a given input by setting weights learned with an unsupervised learning algorithm.

7 Conclusion

To our knowledge we have presented the first free/open-source MT system between Kazakh and Turkish. The performance is similar to other translators created using the same technology, and in

[9] `http://wiki.apertium.org/wiki/apertium-eval-translator`

[10] `https://github.com/moses-smt/mosesdecoder/blob/master/scripts/generic/mteval-v13a.pl`

[11] `https://translate.google.com/#kk/tr/`

[12] `https://ceviri.yandex.com.tr/`

System	OOV	WER (%)	PER (%)	BLEU
Yandex	43	69.73	48.63	2.84
Apertium	128	45.77	41.69	**16.67**
Google	5	**43.85**	**33.67**	16.32

Table 4: Word error rate and Position-independent word error rate; OOV is the number of out-of-vocabulary (unknown) words. The Google system has a similar word error rate to the Apertium system despite the significantly lower number of out-of-vocabulary words. Note that the BLEU scores are computed against a *postedítted* reference translation.

terms of WER to SMT systems available. The system beats the SMT systems on BLEU, while having a much higher out-of-vocabulary rate. This would suggest that given better vocabulary coverage the system would perform significantly better than SMT and NMT systems. The system is available as free/open-source software under the GNU GPL and the whole system may be downloaded from Github.[13]

Acknowledgements

We would like to thank Marzhan Zhakupova for translating the Kazakh sentences in the development corpus.

References

Altenbek, G. and Xiao-long, W. (2010). Kazakh segmentation system of inflectional affixes. In *Proceedings of CIPS-SIGHAN Joint Conference on Chinese Language Processing*, pages 183–190.

Altıntaş, K. (2001). *Turkish to Crimean Tatar machine translation system*. PhD thesis, Bilkent University.

Altintas, K. and Cicekli, I. (2001). A morphological analyser for Crimean Tatar. In *Proceedings of the 10th Turkish Symposium on Artificial Intelligence and Neural Networks (TAINN'2001)*, pages 180–189.

Arnold, D. (2003). Why translation is difficult for computers. *Computers and Translation: A translator's guide, Amsterdam and Philadelphia: John Benjamins*, pages 119–42.

Cöltekin, C. (2010). A freely available morphological analyzer for Turkish. In *LREC*, volume 2, pages 19–28.

Çöltekin, Ç. (2014). A set of open source tools for Turkish natural language processing. In *LREC*, pages 1079–1086.

Forcada, M. L., Ginestí-Rosell, M., Nordfalk, J., O'Regan, J., Ortiz-Rojas, S., Pérez-Ortiz, J. A., Sánchez-Martínez, F., Ramírez-Sánchez, G., and Tyers, F. M. (2011). Apertium: a free/open-source platform for rule-based machine translation. *Machine translation*, 25(2):127–144.

Gilmullin, R. (2008). The Tatar-Turkish machine translation based on the two-level morphological analyzer. *Interactive Systems and Technologies: The Problems of Human-Computer Interaction*, pages 179–186.

Hamzaoglu, I. (1993). *Machine translation from Turkish to other Turkic languages and an implementation for the Azeri language*. PhD thesis, MSc Thesis, Bogazici University, Istanbul.

Kairakbay, B. M. and Zaurbekov, D. L. (2013). Finite state approach to the Kazakh nominal paradigm. In *FSMNLP*, pages 108–112.

Karlsson, F., Voutilainen, A., Heikkilae, J., and Anttila, A. (1995). *Constraint Grammar: a language-independent system for parsing unrestricted text*, volume 4. Walter de Gruyter.

Kessikbayeva, G. and Cicekli, I. (2014). Rule based morphological analyzer of Kazakh language. In *Proceedings of the 2014 Joint Meeting of SIGMORPHON and SIGFSM*, pages 46–54.

Koehn, P. (2010). *Statistical Machine Translation*. Cambridge University Press.

Koehn, P. and Knowles, R. (2017). Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39.

Linden, K., Silfverberg, M., Axelson, E., Hardwick, S., and Pirinen, T. (2011). *HFST–Framework for Compiling and Applying Morphologies*, volume 100 of *Communications in Computer and Information Science*, pages 67–85.

[13] https://github.com/apertium/apertium-kaz-tur

Makhambetov, O., Makazhanov, A., Sabyrgaliyev, I., and Yessenbayev, Z. (2015). Data-driven morphological analysis and disambiguation for Kazakh. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 151–163. Springer.

Muhamedow, R. (2016). *Kazakh: A Comprehensive Grammar*. Routledge: Oxford.

Salimzyanov, I., Washington, J., and Tyers, F. (2013). A free/open-source Kazakh-Tatar machine translation system. *Machine Translation Summit XIV*.

Simons, G. F. and Fennig, C. D. (2018). Ethnologue: Languages of the world. Twenty-first edition. *SIL, Dallas, Texas*.

Sundetova, A., Forcada, M., and Tyers, F. (2015). A free/open-source machine translation system for English to Kazakh. In *3rd International Conference on Turkic Languages Processing (TurkLang 2015), Kazan, Tatarstan*, pages 78–91.

Tantuğ, A. C., Adalı, E., and Oflazer, K. (2006). Computer analysis of the Turkmen language morphology. In *Advances in Natural Language Processing*, pages 186–193. Springer.

Tantuğ, A. C., Adalı, E., and Oflazer, K. (2007a). Machine translation between Turkic languages. pages 189–192. Association for Computational Linguistics.

Tantuğ, A. C., Adalı, E., and Oflazer, K. (2007b). An MT system from Turkmen to Turkish employing finite state and statistical methods.

Tukeyev, U., Melby, A., and Zhumanov Zh, M. (2011). Models and algorithms of translation of the Kazakh language sentences into English language with use of link grammar and the statistical approach. In *Proc. of IV Congress of the Turkic World Math. Society*, pages 1–3.

Tyers, F. M., Sánchez-Martínez, F., Forcada, M. L., et al. (2012a). Flexible finite-state lexical selection for rule-based machine translation.

Tyers, F. M. and Washington, J. (2015). Towards a free/open-source universal dependency treebank for Kazakh. In *Proceedings of the 3rd International Conference on Computer Processing in Turkic Languages (TurkLang)*, pages 276–289.

Tyers, F. M., Washington, J. N., Salimzyanov, I., and Batalov, R. (2012b). A prototype machine translation system for Tatar and Bashkir based on free/open-source components. In *First Workshop on Language Resources and Technologies for Turkic Languages*, page 11.

Washington, J., Ipasov, M., and Tyers, F. M. (2012). A finite-state morphological transducer for Kyrgyz. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC)*, pages 934–940.

Washington, J., Salimzyanov, I., and Tyers, F. M. (2014). Finite-state morphological transducers for three Kypchak languages. In *Proceedings of the 9th Conference on Language Resources and Evaluation (LREC)*, pages 3378–3385.

SRL for low resource languages isn't needed for semantic SMT

Meriem Beloucif and **Dekai Wu**
Human Language Technology Center
Hong Kong University of Science and Technology
Clear Water Bay, Hong Kong
mbeloucif|dekai@cs.ust.hk

Abstract

Previous attempts at injecting semantic frame biases into SMT training for low resource languages failed because either (a) no semantic parser is available for the low resource input language; or (b) the output English language semantic parses excise relevant parts of the alignment space too aggressively. We present the first semantic SMT model to succeed in significantly improving translation quality across many low resource input languages for which no automatic SRL is available —consistently and across all common MT metrics. The results we report are the best by far to date for this type of approach; our analyses suggest that in general, easier approaches toward including semantics in training SMT models may be more feasible than generally assumed even for low resource languages where semantic parsers remain scarce.

While recent proposals to use the crosslingual evaluation metric XMEANT during inversion transduction grammar (ITG) induction are inapplicable to low resource languages that lack semantic parsers, we break the bottleneck via a vastly improved method of biasing ITG induction toward learning more semantically correct alignments using the *monolingual* semantic evaluation metric MEANT. Unlike XMEANT, MEANT requires only a readily-available English (output language) semantic parser. The advances we report here exploit the novel realization that MEANT represents an excellent way to semantically bias expectation-maximization induction even for low resource languages. We test our systems on challenging languages including Amharic, Uyghur, Tigrinya and Oromo. Results show that our model influences the learning towards more semantically correct alignments, leading to better translation quality than both the standard ITG or GIZA++ based SMT training models on different datasets.

1 Introduction

Statistical machine translation (SMT) for low resource languages has been a difficult task due to the unavailability of large parallel corpora. It becomes imperative to make learning from *small* data more efficient by adding additional constraints to create stronger inductive biases—especially linguistically well-motivated constraints, such as the shallow semantic parses of the training sentences. However, while automatic semantic role labeling (SRL) is readily available to produce shallow semantic parses for a high-resource *output* language (typically English), the problem is that SRL is usually not available for low resource *input* languages such as Tigrinya, Oromo, Uyghur or Uzbek.

In this paper, we propose a new method which adopts the monolingual semantic evaluation metric MEANT as a confidence-weighting measure to assess the degree of goodness of training instances, giving a newer strategy than Beloucif and Wu (2016a) who used the degree of compatibility or similarity between the semantic role labeling of the input and output sentences. Their approach might outperform ours for high-resource languages, but is completely inapplicable to low resource languages because XMEANT requires both the input and output semantic parses – whereas MEANT does not require an SRL parse for the low resource input language.

Pérez-Ortiz, Sánchez-Martínez, Esplà-Gomis, Popović, Rico, Martins, Van den Bogaert, Forcada (eds.)
Proceedings of the 21st Annual Conference of the European Association for Machine Translation, p. 59–68
Alacant, Spain, May 2018.