Structural Transfer Rules for Kazakh-to-English Machine Translation in The Free/Open-Source Platform Apertium

Aida Sundetova *Aidana Karibayeva* *Ualsher Tukeyev*
Information Systems Department, Al-Farabi Kazakh National University, Almaty, Kazakhstan
sun27aida@gmail.com; aidana_karib@mail.ru; ualsher.tukeyev@gmail.com;

Abstract

This paper describes process of building structural transfer rules for Kazakh-to-English machine translation system on free/open-source Apertium platform. Structural transfer rules are used for translating texts from Kazakh to English by couple of rules in three stages. This paper shows how sentences in Kazakh are transformed to English sentences, what types of phrases and attributes are used. Results are presented by comparing Apertium Kazakh—English system with other online translators.

1 Introduction

Nowadays developing machine translation from Kazakh language to English is very important and useful for people who want to understand texts in Kazakh and translate them. However, building translation system from a Turkic language, which has complex agglutinative morphology, faces some difficulties. For example, Kazakh morphology, as all Turkic language morphologies, is more complex than English morphology and very different from it. It is impossible to do translation form Kazakh to English by word-to-word. Because Kazakh is agglutinative language, words are done by adding morphemes with vowel harmony (synharmonism) [1]. English is an analytic language that conveys grammatical relationships without using complex inflectional morphemes like in Kazakh language. To be more precise, relationships are expressed by additional constructions with modal verbs or prepositions [2].

There are important differences in syntax between the Kazakh and English languages; for example, the order of constituents in sentences: subject–object–adverbial modifier–verb (in English it is: subject–verb–object–adverbial modifier). There are also important differences in translating verb tenses: Future Simple and Present Simple, Present Perfect and Past Perfect in Kazakh have the same translation, modal verbs are made by adding auxiliary verbs (I can play – Мен ойнай аламын) or using adjectives which mean "obligation": жөн ('should'), қажет ('necessary'), керек ('need') (I should go – Менің барғаным жөн) [3].

Kazakh language has no gender, so personal pronoun "Ол" could has three translations: he/she/it. By default, it is translated as "he", however, for special constructions as "Ол – қыз" (in English "She is girl") "Ол" is translated as "She".

By considering these features, we are developing machine translation from Kazakh to English based on the Apertium free/open-source machine translation platform (Forcada et al. 2011, http://www.apertium.org) [4]. Because, firstly, it already contains a rather complete Kazakh morphology (Salimzyanov et al. 2013), secondly, it includes an English monolingual dictionary which also contains morphological analysis [5]. Therefore for developing Kazakh–English system we need to

Gönderme ve kabul tarihi: 27.09.2014-25.10.2014

build bilingual dictionary and write couple of rules.

This paper contains 4 sections: Section 2 describes Apertium platform and its structure, Section 3 describes Kazakh–English structural transfer and Section 4 gives results of system by comparing with other systems.

2 Apertium platform and its modules

Apertium is a free/open source machine translation system. Apertium is a platform of machine translation which whose development started with financing from the governments of Spain and Catalonia at University of Alicante (Universitat d'Alacant) in 2005. Apertium is free software which is published by developers according to GNU GPL conditions.

Apertium was originally intended for translation between related languages. However this system has been expanded to translate texts between less similar language pairs. To create the new system of machine translation one needs develop linguistic data (dictionaries, rules) in accurately specified XML formats. This system uses finite state transducers for all of its lexical transformations, and hidden Markov models for part-of-speech tagging or word category disambiguation.

Apertium platform consisting of the modules (Figure 1):

– **Deformatter**. It separates the text to be translated from the formatting tags. Formatting tags are encapsulated as "superblanks" that are placed between words in such a way that the remaining modules see them as regular blanks.

– **Morphological analyser**. For each surface form (that is, for each lexical unit as it appears in the text), the morphological analyser generates one or more lexical forms composed of: lemma (dictionary or citation form), lexical category (or part-of-speech), and inflection information. The morphological analyser executes a finite-state transducer generated by compiling a morphological dictionary for the source language. Lexical units containing more than one word (multiword lexical units) are analyzed as a single lexical unit. Morphological analyser uses a finite state transducer based on two-level rules (in the case of Kazakh, `apertium-kaz.kaz.lexc`, `apertium-kaz.kaz.twol`). This module therefore separates lexemes and processes morphological analysis, and then returns possible lexical forms.

– **Part-of-speech (POS) tagger**. Apertium's POS tagger is based on a statistical model based on hidden Markov models which processes the result of the application of on constraint-grammar rules (Karlsson 2005), which are used to discard some analyses using simple rules (written in `apertium-kaz.kaz.rlx`) based on context. For example, consider the morphological analysis of word *қара*:

```
^қара/қара<adj>/қара<adj><advl>/қара<adj>
<subst><nom>/қара<v><tv><imp><p2><sg>/
қара<adj>+e<cop><p3><pl>/қара<adj>+e<co
p><p3><sg>/қара<adj><subst><nom>+e<cop
><p3><pl>/қара<adj><subst><nom>+e<cop>
<p3><sg>
```

This word is ambiguous and has 6 meanings. Many surface forms are ambiguous, which means that these words have more than one POS and therefore more than one possible translation. After this module, all words have only one morphological analysis.

– **Lexical transfer**. This module uses a bilingual dictionary (`apertium-eng-kaz.eng-kaz.dix`) which has very simple structure [7]. The module reads each source-language lexical form and finds one or more corresponding target-language lexical forms. Multiword units are translated as a single word.

– **Lexical selection**. It uses rules that select for those lexical words having many translations, one of the translations in the target language according to context. All rules are written in file `apertium-eng-kaz.kaz-eng.lrx`.

– **Structural transfer.** This module identifies sequences of lexical forms (phrases or segments), which need syntactical

processing (handling of number, prepositions, etc.) to be translated. It uses files with rules, which specify the syntactic transformation as a cascaded process. Transfer rules, which transform lexical-form sequences into a new sequences for the target language, perform the work in this module. Structural transfer is the focus of this paper, and will be described in detail in section 3.

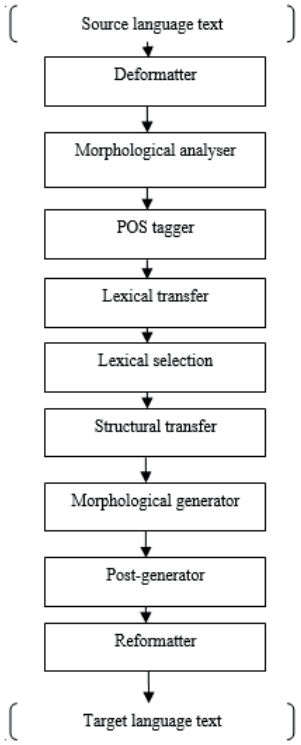

Figure-1. The Apertium machine translation pipeline

– **Morphological generator**. From the sequence of target-language lexical forms produced by the structural transfer, it generates a corresponding sequence of target language surface forms. The morphological generator executes a finite-state transducer generated by compiling a morphological dictionary for the target language.

– **Post-generator**. It takes care of some minor orthographical operations in the target language (for instance, it generates the English form *cannot* from *can* and *not*). This module is generated from file with rules which are very similar in format to dictionary files.

– **Reformatter**. It places format tags back into the text so that its format is preserved.

3 Structural transfer from Kazakh into English languages

The structural transfer module in Apertium does operations, which determined in transfer rules and can be like this: word reordering, adding some suffixes, removing unnecessary tags or attributes etc [8]. Structural transfer in Apertium system comprises two parts: *pattern* and *action*. "*Pattern"* defines the sequence to which the rule will be applied, whereas "*action"* consists of the actual operations needed to generate the corresponding sequence in the target language. Transfer in Apertium may be of two types. The first type is used in a similar languages and generates the sequence of lexical forms in the target language in a single step. The second type is the one used in our Kazakh–English system, and consists of three levels:

– "chunker" level (file `apertium-eng-kaz.kaz-eng.t1x`);

– "interchunk" level (file `apertium-eng-kaz.kaz-eng.t2x`);

– "postchunk" level (file `apertium-eng-kaz.kaz-eng.t3x`).The following sections describe the three levels of Kazakh-English structural transfer.

3.1 The Kazakh-English chunker

The chunker divides a sentence in *chunks* which may be seen as elementary sentence constituents such as noun phrases, verb phrases, etc. (see Table 1)

Table-1. Types of chunks

Patterns	Meaning
SN	Noun phrase
SV	Verb phrase
AdjP	Adjectival phrase
PP	Postpositional phrase

Some examples of noun- and verb-phrase chunks are given in the next tables (Table 2, Table 3):

Table-2. Noun-phrase chunks

Input pattern[1]	Example	Output block	Translation
n	бақша	SN{n}	garden
adj	әдемі	AdjP{adj}	beautiful
num	жеті	SN {num}	seven
adj n	әдемі бақша	SN{adj n}	beautiful garden
det n	менің бақшам	SN{det n}	my garden
num n	жеті бақша	SN{num n}	seven gardens
num adj n	жеті әдемі бақша	SN{num adj n}	seven beautiful gardens
det adj n	менің әдемі бақшам	SN {det adj n}	my beautiful garden
det num adj n	менің жеті әдемі бақшам	SN {det num adj n}	my seven beautiful gardens
n pr	үстел астында	PP {pr n}	under table
adj n pr	үлкен үстел үстінде	PP {pr adj n}	on big table
num n pr	бес үстел үстінде	PP {pr num n}	on five tables

Table-3. Verb-phrase chunks

Input pattern	Example	Output block[2]	Translation
v	ойна	SV{vblex}	play
v	ойнап отыр	SV{vbser vblex }	is playing
v	ойнаған	SV{vbhaver vblex}	has played
v	ойнамаған	SV {vbhaver adv vblex}	has not played
v	ойнар	SV {vaux vbhaver vblex}	will have played

1 Abbreviations: adj, adjective; n, noun; num, numeral; pr, postposition; det, determiner.

2 Abbreviations: vblex, lexical verb; vbser, verb 'to be'; vaux, auxiliary verb; vbhaver, verb 'to have'.

Take into account that the lexical forms have been translated in advance and that the remaining transfer modules work only on target-language lexical forms.

After these blocks (chunks) are created the interchunk module performs operations on these blocks, without modifying their contents. This module makes it possible to generate the correct target-language word order, to treat number and person, number agreement in verbs.

3.1.1 Translation of noun-phrases

We will illustrate the translation of noun-phrase with the example: *әдемі бақшаларда* ('in the beautiful gardens').

The chunker identifies this phrase as a noun-phrase (adj noun) and after that, it translates it into English by adding relevant tags. There may be such tags: number (plural form), cases (assign locative case).

In general, this phrase has the following attributes: number (singular or plural), cases, possessives. One of the main problems in translation noun-phrases is generating the English articles (*a, an, the*), which are absent in Kazakh. All nouns with nominative and accusitive cases are translated as noun-phrases:

- single noun: SN [қыз <n><nominative>] - SN [girl <n><nominative><sg>];

Also for structue like:

- adjective + noun: SN [әдемі<adj> үй<n><nom> - SN [beautiful<adj> house<n><sg>];
- numerals + noun (in accusative case): SN [жеті<num> бақша<n><accusative> - SN [seven<num> garden<n><plural>]. Rules for this phrase are not assigned to noun accusative case because in English translation it does not have any suffixes.

3.1.2 Translation of verb phrases

Translation of verb from Kazakh to English has specific difficulties. For instance, in

Kazakh the past tense might have two translations; for example, the sentence “Мен ойнағанмын” can be translated such as “I have played” or “I had played”, that is, the sentence can be translated as present perfect or past perfect. We decided to generate a present perfect translation, because in while developing in first steps it difficult to identify past perfect, which has to come before past simple and present perfect are more common in simple sentences. Below are shown examples of verb-phrases, which the system already translates (Table 4):

Table-4. Translation of verb phrases

Tense in Kazakh language	**Example**	**Tense in English**	**Translation**
Present (Ауыспалы осы шақ)	*Мен ойна+й+мын*	Present Simple	*I play*
Past (Жедел өткен шақ)	*Мен ойна+дым*	Past Simple	*I played*
Future (Болжалды келер шақ)	*Мен ойна+р+мын*	Future Perfect	*I will have played*
Present (Нақ осы шақ)	*Мен ойна+п жатыр+мын*	Present Continuous	*I am playing*

3.1.3 Translation of adjectival and postpositional phrases

Adjectival phrases do not have any attributes and are marked as “AdjP”, and are used for those cases in which adjectives are not part of a noun phrase.

Postpositional phrases are structures in which function words are found after the noun. For instance, the phrase «жеті әдемі бақшаның астында» translated as «under seven beautiful gardens». In a construction like this the compound postposition formed by the genitive ending “*-ның”* in “*бақшаның”* and the word “астында*”* are used to express the notion expressed in English with the function word “under”.

In this level of transfer rules are written 57 rules.

3.2 “Interchunk” level

As we can see from the other translation systems (try translate texts [9,10]), in target-language texts word order is incorrect. It means that reordering does not work well. When we write a chunker rule (`.t1x`), we aim at dividing the sentence in a sequence of patterns or chunks. After that, we take care of the order of these chunks by writing interchunk rules in file `apertium-eng-kaz.kaz-eng.t2x`, by writing appropriate reordering rules. For instance, in sentence “Біз кітапты оқимыз” pattern of pronoun “Біз” ('We') is “SN”, pattern of object “кітапты” ('book') is “SN-accusative” and pattern of verb “оқимыз” ('read') will be “SV”: “Біз кітапты оқимыз” - “We read book”(reordering “We book read”). So in Kazakh language verb stays at the end of sentence, although in English that can stay at the beginning or in the middle: “Мен[1] әдемі[2] бақшаны[3] көремін[4]” → “I[1] see[4] beautiful[2] garden[3]”. Rules of this level do next operations:

- build new sequence of chunks;
- adding prepositions by cases: “әдемі бақшаДА[locative]” - “**in** beautiful garden”;
- agreement. Agreement of words – subject and verb, adjective and noun, for example, for agreement between verb and subject are person and number agreement. For example: “*Бала ойнайды” – “Child plays”* (noun is third person and number is singular, so why noun should have morpheme of person).

The number of rules like this is some few, about ten rules, furthermore these rules will be extended.

4 Results

The current version of the system (revision №56387) can translate SN-, SV-. AdjP- and PP- phrases. We plan to extend the number of rules to improve translation quality. In the table below we compare some translation systems with examples that our system can translate [9, 10]. All available translations of sentences and phrases can be seen from tests (see [11]).

Table-5. Results of comparison

Phrase	Example	Apertium	Pragma 6	Sanasoft
SN	менің екі әдемі көйлегім	My two beautiful dresses	*me* two *әдемі* my the dress	My two beautiful *көйлегім*
SV	Мен студент емеспін	I am not student	I *not* student	I student *емеспін*
PP	Анау суреттер дің астында	under those pictures	Under those *by* pictures	under *that* pictures

5 Conclusion

We have described Kazakh—English machine translation system on Apertium platform and process of developing structural transfer rules. Many features in translating from Kazakh to English as assign cases, agreement, prepositions, etc. were solved. In the future this system will be considered the translation task of future transitional tense, the passive voice, degree adjective, interrogative sentence and other tasks will be observed.

Acknowledgements: the authors thank Mikel L. Forcada, Francis Tyers, Jonathan N. Washington and Ilnar Salimzyanov and other developers in the Apertium project for their help during the development of this system, authors would also like to express their gratitude to Mikel L. Forcada for advises in writing this paper.

6 References

[1] Агглютинативные языки (2012). Retrieved from http://ru.wikipedia.org/wiki/Агглютинативный_язык

[2] Аналитический язык (2013). Retrieved from http://ru.wikipedia.org/wiki/Аналитический_язык

[3] Печерских, Т.Ф., Амангельдина, Г.А. (2012) “Особенности перевода разносистемных языков (на примере английского и казахского языков)”, Молодой ученый. №3, 259–261

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

[5] Salimzyanov, I., Washington, J.N., Tyers, F.M. “A free/open-source Kazakh-Tatar machine translation”. Proceedings of MT Summit XIV (Nice, France, 4–6 September 2013), accepted.

[6] Karlsson, F., Voutilainen, A., Heikkilä, J., Anttila, A. 1995. Constraint Grammar: A Language-Independent System for Parsing Unrestricted Text. Mouton de Gruyter, Berlin.

[7] Сундетова А.М., Кәрібаева А.С., Апертиум платформасындағы Ағылшын–Қазақ машиналық аудармашы үшін екітілді сөздікті құру. Материалы международной научно-практической конференции «Применение информационно-коммуникационных технологий в образовании и науке», посвященной 50-летию Департамента информационно-коммуникационных технологий и 40-летию кафедры «Информационные системы» КазНУ им. аль-Фараби. 22 ноября 2013г. – Алматы: Қазақ Университеті, 2013. – С.53-57.

[8] Sundetova A., M.L. Forcada, A. Shormakova, A.Aitkulova, Structural transfer rules for English-to-Kazakh machine translation in the free/open-source platform Apertium. Компьютерная обработка тюркских языков. Первая международная конференция: Труды. – Астана: ЕНУ им. Л.Н. Гумилева, 2013. – С. 317-326.

[9] Online-translator «Sanasoft»: http://www.sanasoft.kz/c/ru/node/47 (in Russian), http://www.sanasoft.kz/c/kk/node/53 (in Kazakh).

[10] Online-translator «Trident: http://www.translate.ua/us/on-line;

[11] Regression tests. http://wiki.apertium.org/wiki/English_and_Kazakh/Regression_tests