Technology of the structural machine translation rules generation, based on the complete set of Kazakh endings

Ualsher Tukeyev, Aida Sundetova, Balzhan Abduali, Aidana Karibayeva, Dina Amirova

Al-Farabi Kazakh National University, Information Systems department,
Al-Farabi av., 71, 050040 Almaty, Kazakhstan
ualsher.tukeyev@gmail.com, sun27aida@gmail.com,
balzhanabdualy@gmail.com, a.s.karibayeva@gmail.com,
amirovatdina@gmail.com
http://www.kaznu.kz

Abstract. In this paper we propose technology of generating a set of machine translation rules, based on the system of the complete set of Kazakh endings. The proposed technology is based on the combination of using a complete set of endings types of the rich morphology source language, such as the Kazakh language, free/open-source Apertium platform and rule generation application. Structural transfer rules extraction is shown for English-Kazakh and Kazakh-Russian machine translations systems.

Keywords: inferring, Apertium platform, machine translation, chunk, transfer rules, endings.

1 Introduction

Kazakh language, as one of Turkic languages, belongs to an agglutinative language, and it uses vowel harmony. Which means that translating text in Kazakh language into other languages with simpler morphology, such as English or Russian languages, with, for instance, statistical machine translation(SMT), will cause some quality loss because of morphological segmentation.

In previous work [1] we already built application to extract rules from complete set of endings for Kazakh language, where application used only prepared, by hand, template with morphological analysis. In this paper we propose full technology of extracting rules from the original phrases, based on the system of complete set of endings. Phrases will be processed by Apertium platform's morphological analyzer and rules extracting application. All examples will be shown for English-Kazakh and Kazakh-Russian machine translation systems.

2 Problem of creating rules for Kazakh language

The problem of creating rules is hidden in the methods of their building. In rule-based machine translation (RBMT) systems usually rules are created by humans, which could cause some questions, such as: is set of rules, created by humans, complete or full? In case of SMT, "rules" to translate are applies by considering probabilities of the translations and in fact, it does not guarantee that in particular case phrase with low probability will be translated correctly [2].

In this paper we propose our solution of the problem: usage of complete set of Kazakh endings, which gurantees that rules, are built from this system, will be also complete and technology, where human's work will be less than in RBMT, because of automatical rule generalisation. In the next section 3 we will show how complete set of endings was created, in section 4 improved process of rule extraction will be described, in section 5 some experiments and program realisation will be shown.

3 Related works

In the previous work we considered how to wrap all the endings of one language and by automatic system fill in all the rules that you can write for all of these endings. In the previous work, the chunk transfer rule logical templates for Kazakh-English and Kazakh-Russian words with nominal base were constructed.

The question of automatic inferring of the structural rules of machine translation from one language to another are rather actual for machine translation systems based on grammatical rules (RBMT). This is due to the time-consuming process of drawing up the rules for RBMT.

The scheme of inference transfer rules based on the following: the source is a complete set of endings types of the Kazakh language; for each types of endings (template of morphological structure types of endings) of the Kazakh language is constructed a equivalent grammatical structure logical template (pattern) in the target language (for example, Russian and English languages); on the basis of grammatical structure logical template built template of program structure for transfer of the morphological structure of word’s endings into the equivalent grammatical structure of the target language [1].

In this paper proposed the technology of extracting rules from complete set of Kazakh endings and example of how can work this system. This approach allows generating a complete system of morphological chunk transfer rules.

4 Technology of extracting rules from complete set of Kazakh endings

As was described in the previous section, application for generalization of rules, based on the complete set of Kazakh endings, used templates in the following format:

<n><pl><nom>|<n><m><nn><pl><nom>

Where left side is source language and on the right side is target language phrase's morphological analysis.

This template has some disadvantages:

- To get morphological analysis user has to use additional application;
- Template should be build by hand;
- Only one phrase could be used in one time.

To fix these issues new technology of extracting rules was proposed: to do morphological analysis will be used a free/open-source machine translation platform Apertium. There are several reasons:

- Apertium is open source, which means that code of this project could be used or modified to insert to previous extraction application.
- This project has morphological analyzer for Kazakh, Russian and English languages, monolingual resources, such like English, Kazakh, Russian dictionaries would be used in process to get template.

New workflow of extracting rules, based on the next schema:

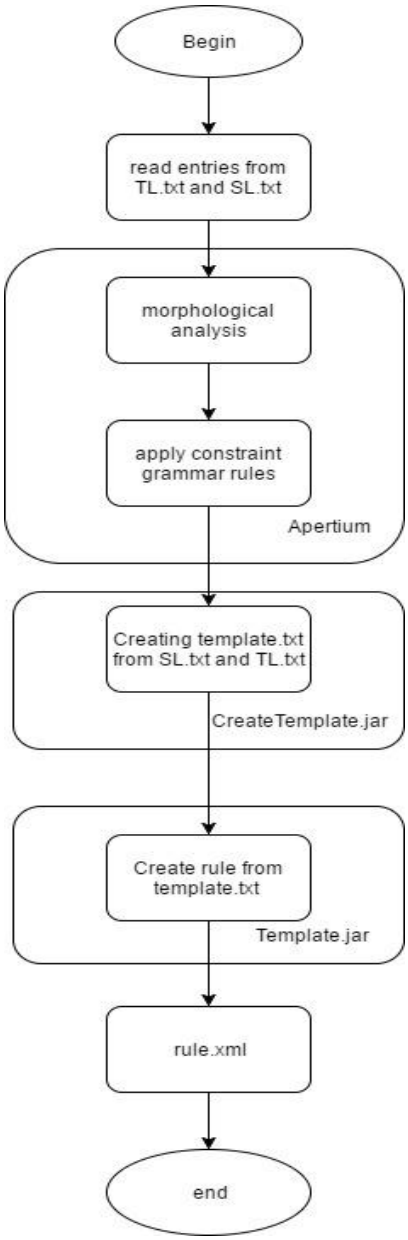

Fig. 1. New workflow of extracting rules

As can be seen from the schema, workflow includes several steps:

- Two files are created: first file, which contains part of phrase in source language and in translation is of phrase is put to second file.

- Apertium platform is used to get morphological analysis of phrases from created files.
- Morphological ambiguity is also solved with Apertium instruments, such like constraint grammar module [3]. If phrase is translated from English, additionally part-of-speech tagger, based on the hidden Markov model will be used [4].
- To create template from both analysis, simple script(CreateTemplate.jar) was built. This script reads two analysis from source and target language phrase translations and creates another file with name of rule to be created and on the second line template itself, for instance:

```
S-K
<n><pl><nom>|<n><m><nn><pl><nom>
```

- Finally, application(Template.jar), which will generate the rule, reads name of chunk and template from "template" file.
- All process ends with generating one file in .xml format containing rule, which could be used in transfer stage of Apertium platform.

In the section 5 will be shown few results of the this technology realization.

5 Example of program realization

To develop workflow, described above, was used Apertium platform with installed apertium-kaz [5], apertium-rus [6], apertium-eng-kaz [7] machine translation systems, application to create templates and application to extract rules from template itself. For instance, let's consider that we are going to build rule for plural form of nouns, which chunk is named as "S-K". Firstly, as was mentioned above, user has to prepare two files: file, containing phrase in source language(SL.txt) and file with translation of this phrase in target language (TL.txt). Then, content of SL.txt will be:

Table 1. Example of two files for build chunk of “S-K“

SL.txt	*TL.txt*
in garden	бақшада
in school	мектепте

Afterwards, user will be start script to extract rules, which starts all the process described above. In the result, rule.xml file will be created. Below is shown piece of this file:

```
  <section-rules>
<rule comment="RULE:S-K">
<pattern>
 <pattern-item n="cat_pr"/>
 <pattern-item n="cat_n"/>
</pattern>
  <action>
 <out>
      <chunk name="__S-K__" case="caseFirstWord">
     <tags><tag><lit-tag v="NP"/></tag>       </tags>
  <lu>
  <clip pos="2" side="tl" part="lem"/>
  <clip pos="2" side="tl" part="_attr_n_"/>
<lit-tag v="loc"/>
</lu>
<b/>  </chunk>
  </out>
 </action>
 </rule>
 </section-rules>
</transfer>
```

Fig. 2. The result, rule.xml file

This rule could be easily integrated into first stage of structural transfer of Apertium – chunker level [8], because in the generating script additionally user should add name of pharse, for instance, noun phrase – NP. Below we show the how rule translated the structure "in garden - бақшада(baqshada):

```
aida@aida-HP-Pavilion-Notebook:~/apertium-testing/apertium-eng-kaz$ echo "in gar
den" |apertium -d. eng-kaz-transfer

apertium-transfer: Rule 1 in<pr>/ garden<n><sg>/бақша<n><sg>
^__S-K__<NP>{^бақша<n><loc>$ }$^default<default>{^.<sent>$}$
aida@aida-HP-Pavilion-Notebook:~/apertium-testing/apertium-eng-kaz$ echo "in gar
den" |apertium -d. eng-kaz
бақшада
aida@aida-HP-Pavilion-Notebook:~/apertium-testing/apertium-eng-kaz$
```

Fig. 3. Translating the phrase “in garden - бақшада(baqshada”

We did experiment for Kazakh-Russian language pair with structure "S-K-T-C", where source language is Russian. Below, in figure 3 could be seen translation with original Apertium Kaz-Rus [9]:

```
aida@aida-HP-Pavilion-Notebook:~/apertium-testing/apertium-kaz-rus$ echo "к моим
 тётям"|apertium -d. rus-kaz
 #мен тәтелерімізге
aida@aida-HP-Pavilion-Notebook:~/apertium-testing/apertium-kaz-rus$
```

Fig. 4. Experiment for Kazakh-Russian language pair with structure “S-K-T-C”

By using complete set of the Kazakh settings, we were able to identify, that rule for structure "S-K-T-C" does not exist in Apertium-kaz-rus machine translation system. In files named "rus.txt" and "kaz.txt" were put structures to generate rule:

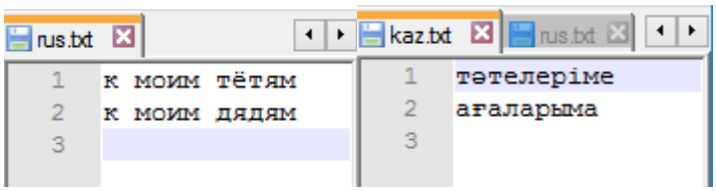

Fig. 5. The structure “S-K-T-C” to generate rule

After running generation script we got the rule:

```
<rule comment="RULE:S-K-T-C">
<pattern>
 <pattern-item n="cat_pr"/>
 <pattern-item n="cat_prn"/>
 <pattern-item n="cat_n"/>

</pattern>
  <action>
 <out>
       <chunk name="__S-K-T-C__" case="caseFirstWord">
      <tags><tag><lit-tag v="NP"/></tag>       </tags>
  <lu>
  <clip pos="3" side="tl" part="lem"/>
<clip pos="3" side="tl" part="_attr_n_"/>
<lit-tag v="px1sg"/>
<lit-tag v="dat"/>
</lu>
<b/>  </chunk>
  </out>
 </action>
 </rule>
 </section-rules>
</transfer>
```

Fig. 6. The rule for structure “S-K-T-C” that were taken from generating

The translation could be seen on the following picture:

```
aida@aida-HP-Pavilion-Notebook:~/apertium-testing/apertium-kaz-rus$ echo "к моим
 тётям"|apertium -d. rus-kaz
тәтелеріме
aida@aida-HP-Pavilion-Notebook:~/apertium-testing/apertium-kaz-rus$ echo "к моим
 тётям"|apertium -d. rus-kaz-transfer
^__S-K-T-C__<NP>{^тәте<n><pl><px1sg><dat>$ }$^default<default>{^.<sent>$}$
aida@aida-HP-Pavilion-Notebook:~/apertium-testing/apertium-kaz-rus$
```

Fig. 7. Translation the structure “S-K-T-C”

As could be seen from the screenshot, generated rule translates without any error, as "#" sign, it means that rule is generated correctly and could work with other Apertium platform transfer's stage.

6 Conclusion and future work

In this paper we performed impoved technology of construction the chunk transfer rules, based on the complete set of Kazakh endings, for Kazakh-Russian and Kazakh-English language pairs. Technology, beside using previously developed application to generate rules, is used morphological analyzer of Apertium platform, which helps to do process of rule generation more easily.

In the future work is planned to improve rule generation program to use for verb phrases and structures. Now generation application could be applied for noun phrases only.

References

1. Tukeyev U., Sundetova A., Abduali B., Akhmadiyeva Zh., Zhanbussunov N. Inferring of the morphological chunk transfer rules on the base of complete set of Kazakh endings // Proceedinds of 8th International conference ICCCI 2016. – Halkidiki, Greece, 2016. – P. 563-574.
2. Philipp Koehn «Statistical Machine Translation». Cambridge University Press The Edinburgh Building, Cambridge CB2 8RU, UK. Published in the United States of America by Cambridge University Press, New York. ISBN-13 978-0-521-87415-1 Hardback.
3. Karlsson F., Voutilainen A., Heikkilä J., Anttila, A. Constraint Grammar: A language independent system for parsing unrestricted text. //Mouton de Gruyter. – 1995.
4. Doug Cutting, Julian Kupiec, Jan Pedersen, and Penelope Sibun. A Practical Part-ofspeech Tagger. Proceedmgs of ANLP-92. Trento, Italy, 1992.
5. https://svn.code.sf.net/p/apertium/svn/languages/apertium-kaz
6. https://svn.code.sf.net/p/apertium/svn/languages/apertium-rus
7. https://svn.code.sf.net/p/apertium/svn/staging/apertium-eng-kaz
8. Sundetova, A., Forcada, M. L., Shormakova, A., and Aitkulova, A. (2013). Structural transfer rules for english-to-kazakh machine translation in the free/open-source platform apertium. In Proceedings of the International Conference on Computer processing of Turkic Languages, pages 317–326.
9. https://svn.code.sf.net/p/apertium/svn/staging/apertium-kaz-rus/