Machine Translation of Different Languages with reference to English and Kazakh languages

Shormakova Assem, Sundetova Aida
Department of Information Systems
Kazakh National University,KazNU
Almaty, Kazakhstan
assem007@mail.ru, sun27aida@gmail.com

ABSTRACT: *This paper describes the first steps in the project of building a prototype of a free/open-source rule-based machine translation system that translates from English to Kazakh, which is based on the free/open-source Apertium machine translation platform. The choice of a free/open-source model is motivated by the objective of providing a free/open-source English to Kazakh system that could be used and modified by anyone interested, and uses, unlike other systems available for this language pair, a well-documented rule- and dictionary-based translation procedure. After a survey of the first problems of English -Kazakh translation tackled, a description follows of: (a) the Apertium platform and the use of the Helsinki finite-state toolkit to model Kazakh morphology, (b) the methodology and the initial resources used to build the linguistic data needed (English and Kazakh monolingual dictionaries, bilingual dictionaries, structural transfer rules), (c) the current status of the resulting system, (d) its availability as free/open-source software, (e) a list of immediate steps to take, and (f) a description of long-term development planned, which includes the use of the system in computer-aided and interactive translation environments for professional translators. A preliminary comparison to two existing commercial English–Kazakh systems, using examples covering the linguistic phenomena for which structural transfer rules have been written (processing of short noun phrases, verb phrases and adpositional phrases) is also provided; the comparison shows that a principled rule-based approach may be expected to easily improve the results of existing commercial systems for English-Kazakh and motivates the effort for future development. The goal of this article is to examine a grammatical and lexical problems, which we often face while translating English texts, and not giving any detailed statement of grammatical or lexical pheno-menon. Apertium platform translates many different languages and it uses Hidden Markov Models. This article shows description of machine translation which translates English to Kazakh.*

Keywords: English, Kazakh, Machine Translation, Rule-based, Free/Open-source Software, Apertium

Received: 1 July 2014, Revised 8 August 2014, Accepted 11 August 2014

© 2014 DLINE. All Rights Reserved

1. Introduction

1.1 Main Differences Between English and Kazakh and Their Relevance for Machine Translation

Kazakh is a Turkic language, and its syntax is very similar to that of Azerbaijani or Turkish (they do not belong to the Kipchak family as Kazakh, but are currently supported, for example, by Google Translate,[1] which is basically a statistical machine translation system). For a quick summary of some of its divergences with English we can mention that:

• it has clitic postpositions (sometimes described as cases) and self-standing postpositions instead of prepositions:

(1) in the garden
бақшада
garden-LOC

(2) under the garden
бақшаның астында
garden-GEN bottom-LOC

• it usually places modifiers before the constituents they modify

(3) Chief of staff
персоналдың көсемі
staff-GEN staff-its

(4) The professor's garden
профессордың бақшасы
professor-GEN garden-his

(note the characteristic double marking of possession using the genitive case or postposition -NIң[2] and the possessives -*i* and *сы* in the second noun).

• finally, it usually places the verb at the end of the sentence:

(5) They played in the garden
Олар бақшада ойнады
They garden-LOC played

As Apertium's capabilities to deal with long-range reordering are limited (see section 1.3), the initial work in this prototype aimed at generating adequate translations for selected short segments (1–5 words) rather than for the whole sentence, much as it is done in current Apertium pairs such as Basque–Spanish (Ginestí-Rosell et al. 2009) or Basque–English, where Basque is syntactically quite similar to Turkic languages.

1.2 Existing systems for the English–Kazakh Language pair

There are only a few machine translation systems that translate from English to Kazakh. We have only been able to find two of them online. The Kazakh company Sanasoft commercializes two systems called Master Word and WinTranslate; their systems can also be used online[3]. An Ukrainian company, Trident, commercializes Pragma, which is available from the company's site,[4] and also available through the iTranslate4.eu portal.[5] We have not found details on how these systems work internally, as the

[1] http://translate.google.com

[2] As usual, capitals are used to represent archi-phonemes that will be realized according to morphotactical rules, in this case *ОЫҢ*.

[3] http://www.sanasoft.kz:8889/Main

[4] http://www.translate.ua/us/on-line

[5] http://itranslate4.eu/en/

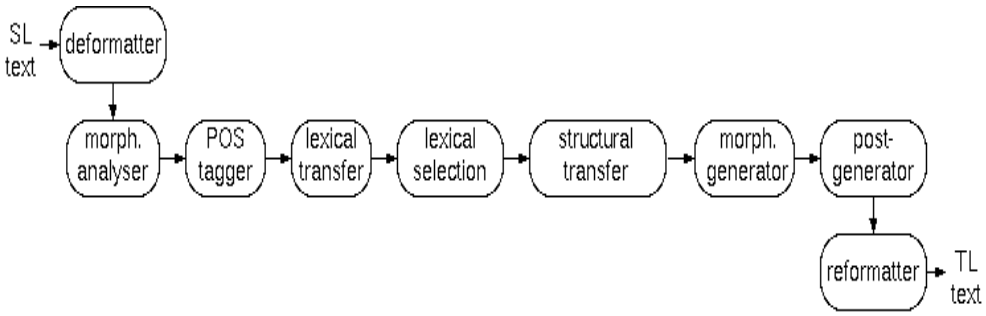
companies have not published much detail.

With the Apertium system under construction we aim at providing a free/open-source English to Kazakh system that could be used and modified by anyone interested, and one that uses a well-documented rule- and dictionary-based translation procedure, which is not the case with either the Sanasoft or the Trident systems.

1.3 The Apertium Free/Open-Source Machine Translation Platform

Apertium[6] (Forcada et al. 2011) is a free/open-source rule-based machine translation platform that was launched in 2005 by the Universitat d'Alacant. Though it was initially aimed at translating between closely related languages, it was also extended to be able to deal with unrelated languages. All of the components of the platform (machine translation engine, developer's tools, and linguistic data for an increasing number of language pairs) are licensed under the free/open-source GNU General Public License (GPL, versions 2 and 3) and are available to everyone interested in the website.

Apertium-based machine translation systems are transfer systems implemented as text pipelines consisting of the following modules:



1 A deformatter that 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.

2 A morphological analyser, yielding, for each surface form (SF), for each lexical unit as it appears in the text, a lexical form (LF) composed of: lemma (dictionary or citation form), lexical category (or "*part-of-speech*"), and inflection information. For instance, the English SF *books* would yield two LFs: (*book*, noun, plural) or (*book*, verb, 3rd person present tense). The morphological analyser executes a finite-state transducer generated by compiling a *morphological dictionary* for the source language (SL).

3 A constraint-grammar (Karlsson 2005) module based on CG3[7] may optionally be used to discard some LFs using simple rules based on context (this module is not depicted in the figure).

4 A part-of-speech tagger based on hidden Markov models (Cutting et al. 1992) selects one of the LFs. The statistical models may be supervisedly trained on an annotated SL monolingual text corpus, or trained in a supervised way, either on an unannotated monolingual SL corpus or using two unrelated, unannotated source language and target language corpora (as in Sánchez-Martínez et al. 2008). The Apertium part-of-speech tagger can also read linguistically-motivated constraints (much more rudimentary than constraint grammar rules in the previous module) that forbid specific sequences of two LFs.

5 A lexical transfer module adds, to each source language LF (SL LF), one or more corresponding target language LFs (TL LFs). This module executes a finite-state transducer generated by compiling a bilingual SL–TL dictionary.

6 An (optional) lexical selection module reads in rules that allow for the selection of one of the TL LFs according to context. When this module is absent, the TL LF given as default in the dictionaries is used.

7 A structural transfer module processes the stream of SL LF–TL LF pairs and transforms it into a new sequence of TL LFs after a series of structural transfer operations: reordering, elimination or insertion of LFs, agreement, etc. Structural transfer rules have a pattern–action form: when a certain pattern of SL LFs is detected, an action generates the corresponding sequence

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

[7] http://beta.visl.sdu.dk/cg3.html

of TL LFs. Rules are applied in a greedy left-to-right, longest-match fashion. There are two main modalities of structural transfer. The first one (used for related languages) generates the TL LF sequence in a single step. The second one (used in the English–Kazakh system described in this paper) uses three stages[8] to improve the granularity of structural transfer rules (each one has its own rules file):

a A first round of transformations ("*chunker*") detects SL LF patterns and generates the corresponding sequences of TL LFs grouped in *chunks* representing simple constituents such as noun phrases, prepositional phrases, etc. These chunks bear tags that may be used for further processing.

b The second round ("*interchunk*") reads patterns of chunks and produces a new sequence of chunks. This is the module where one can attempt to perform some long-range reordering operations.

c The third round ("*postchunk*") removes all grouping information to generate the desired sequence of TL LFs.

8 A morphological generator takes the sequence of TL LFs and generates a corresponding sequence of TL SFs. The morphological generator executes a finite-state transducer generated by compiling a *morphological dictionary* for the TL.

9 A post-generator takes care of some minor orthographical operations such as apostrophations and contractions in the target language (not used for English to Kazakh).

10 Finally, the deformatter places the formatting tags back into the text so that its format is preserved.

2. Methodology

2.1 Methodology Used To Build The Data For The Prototype

2.1.1. Initial Ata

As it is common in the Apertium project, building linguistic data for a new language pair may involve the reuse of existing data in other language pairs in the project. The initial data used for the English–Kazakh prototype were:

- A rather complete English morphological dictionary (to be used by the morphological analyser), taken from the stable English–Spanish language pair in Apertium.
- The English part-of-speech tagger of the English–Spanish language pair in Apertium.
- An English constraint grammar (CG3) to avoid some ambiguities found in a small text.[9]
- A skeleton for the English–Kazakh bilingual dictionary containing a single entry.
- A skeleton for the English–Kazakh structural transfer .t1x rule file (*chunker*, see section 1.3) containing one rule for bare nouns and a rule for the end-of-sentence marker
- Skeletons for the level 2 and 3 structural transfer files, containing only the "*default*" rule.
- A prototype of the Kazakh morphological dictionary (to be used by the morphological generator of this system), with about two hundred entries.[10]

This set of initial files were kindly prepared by Francis M. Tyers from existing material in Apertium (package apertium-kaz, used for a Kazakh to Crimean Tatar translator, and package apertium-en-es, the English to Spanish translator).

As regards the engine, a hybrid approach, combining standard Apertium components with external components, was taken to choose its configuration. In particular, the Helsinki finite state toolkit (HFST, Lindén, Silfverberg and Pirinen 2009) was used for morphological generation, as a short HFST dictionary was already available for Kazakh and in view that this is

[8] There are in Apertium pairs that use an experimental modality with more than three stages, but they have to use workarounds as the current architecture was not intended for that.

[9] http://apertium.svn.sf.net/svnroot/apertium/incubator/apertium-eng-kaz/texts/eng.txt

[10] Built by Apertium developer Ilnar Selimcan with help from Apertium developers Jonathan North Washington and Francis M. Tyers.

The sentence	Analysis
I	^Мен<prn><pers><p1><sg><nom>$
eight	^сегіз<num>$
beautiful	^әдемі<adj>$
gardens	^бақша<n><nom>$
through	^арқылы<post>$
was coming	^кел<v><iv><prc_perf>$
	^отыр<vaux><ifi><p1><sg>$

the usual choice when dealing with Turkic languages in the Apertium project.

2.1.2. Building of new data

A mixture of strategies has been used to quickly build data:

• Lexical data (mainly bilingual dictionary and Kazakh monolingual dictionary data, as the English dictionaries are quite complete) have been added initially by adding the vocabulary from a short story[11] and later by adding vocabulary from Ogden's BASIC English.[12]

• As regards the first round structural transfer (*chunker*, .t1x file), rules were written by hand to gradually address the main morphosyntactic divergences between the languages involved. We did not even attempt to address in this prototype the full set of morphosyntactic divergences between English and Kazakh; the initial goal was to get the prototype to perform the local operations necessary to adequately process short noun phrases, verb phrases and *adpositional* phrases (that is, prepositional phrases in English and postpositional phrases in Kazakh); rules were also added to be able to perform basic verb transfer (tense selection, etc.).

• Two example *interchunk* (.t2x) rules were written to place the verb at the end of short sentences by moving around postpositional phrases, such as

(6) He was playing on top of three beautiful trees

Ол үш әдемі ағаштың үстінде ойнап отырды

He three beautiful tree-GEN top-their-LOC playing stayed

During the development of rules, *regression testing w*as used to ensure that the new rules did not break existing correct translations. To that end, a shell script (regression-tests.sh) is included with the package; this shell compares the translations produced by the current revision of the system with those maintained at http://wiki.apertium.org/wiki/English_and_Kazakh/Regression_tests.

2.1.3 Availability

Currently the language pair is being developed in the Sourceforge space assigned to Apertium, in particular

http://apertium.svn.sf.net/svnroot/apertium/incubator/apertium-eng-kaz/ (browsable via http://apertium.svn.sf.net/viewvc/apertium/incubator/apertium-eng-kaz/). Installation instructions for a Debian GNU/Linux-based system may be found in http://wiki.apertium.org/wiki/English_and_Kazakh#Installing_what_is_needed . A prototype which is regularly updated to reflect the current status of development of the package is also provided at http://elx.dlsi.ua.es/~fran/eng-kaz/.

3. Description of the Current System

What follows is a brief description of the English–Kazakh linguistic data available on July 13, 2012 (Subversion revision No. 39404).

[11]http://apertium.svn.sf.net/svnroot/apertium/incubator/apertium-eng-kaz/texts/eng.txt

[12]http://ogden.basic-english.org/

3.1 Description

3.1.1 Dictionaries

The current dictionaries:

- The Kazakh morphological dictionary used for generation contains about 1000 entries.
- The English dictionary has only slightly been updated from the one initially used.
- The bilingual English–Kazakh dictionary (built from scratch) contains about 970 entries.

3.1.2 Transfer Rules

The following shows the current status of the structural transfer:

Constituent type	Transfer rules (t1x)	Interchunk rules (t2x)	Post-chunk rules (t3x)
Noun phrases	8[13]	0	0
Adpositional phrases	8[14]	0	0
Verb phrases	3[15]	2	0
Other	2	1	1

3.2 Immediate Work to be Done

Here is a description of work that is planned to be immediately performed:

3.2.1 Dictionaries

- Complete the addition of vocabulary to the bilingual and Kazakh dictionaries from Ogden's BASIC English, to ensure a reasonable general-purpose coverage of English.
- Improve the handling of closed-class words such as demonstratives, possessives, etc.
- Add support for new clitics such as the -сіз/-сыз 'without' privative or abessive.
- Correcting some problems in the Kazakh HFST morphology, such as the орн/орын root alternation problem.

3.2.2 Structural Transfer Rules

- Rationalize and unify the current naming of macros and variables
- Rationalize the current naming of chunks produced by the chunker for easier interchunk processing.
- Extend current support for more complex noun phrases and adpositional phrases.
- Complete support for verb tense.
- Add "*prep pron*" rules like "*with me*", "*under me*"
- Add one-word chunks in the .t1x file to translate correctly "*loose*" words that have not been covered by any pattern, and therefore, not been processed by any chunk such as adverbs.
- Add t2x rules to reorder (where necessary) NP GenP sequences, and propagate the possessive person of the GenP to the NP.

---

[13] English patterns: noun, determiner–noun, adjective–noun, determiner–adjective–noun, numeral–noun, determiner–numeral–noun, numeral–adjective–noun, determiner–numeral–adjective–noun

[14] All the noun phrases preceded by a preposition.

[15] Simple verb tenses, *be* + verb *-ing* verb tenses and *will* + verb tenses.

3.3 Preliminary Comparison

We have compared the performance of the system being developed to that of the commercial systems available for the set of translation operations that have been addressed (namely, noun phrase and adpositional phrase processing, verb phrase reordering). Here are just a few examples of adequate translation produced by Apertium that are not adequately translated by the commercial systems.

Original	Apertium	Sanasoft	Trident
he was playing on top of three beautiful trees in the new garden	ол жаңа бақшада үш әдемі ағаштың үстінде ойнап отырды.	ол *зырылда уық *three әдемі *ағаштар жаңа *бақ ойнады	ол жаңа бақшада *в *алқындыр- үш тамаша тал- *шыбықтардың ойнады

In the table, * marks an inadequate translation and ? marks a partially inadequate translation.

The results indicate clearly that syntactic processing in the system, based on well defined rules for local and longer-term transformations, appears to be superior than the one found in the commercially-available English–Kazakh systems Sanasoft and Trident.

4. Concluding Remarks and Future Work

This paper has described the initial steps to build the prototype of a free/open-source rule-based English–Kazakh machine translation system based on the Apertium platform (apertium-eng-kaz). The current prototype already successfully solves many cases of noun-phrase and adpositional-phrase translation (actually better than the available commercial systems), and contains a reasonable vocabulary for testing purposes, which nevertheless has been completed.

In addition to the immediate actions listed in section 3.2, here is a longer-range set of tasks to be performed in order to have a working machine translation system:

- Completing the coverage of structural transfer rules and monolingual and bilingual vocabularies so that the system produces a translation for at least 90% of the English words and performs the basic operations to identify and process correctly short constituents (1–6 words).
- Releasing the resulting stable system as apertium-eng-kaz version 0.1 and disseminate it to the interested parties to obtain feedback about its functioning. We can reasonably expect this system to work better than the existing commercial systems in most aspects.
- Inserting the resulting system into computer-assisted translation (CAT) workflows being developed at the Universitat d'Alacant: its use to provide edit hints in translation-memory-based CAT[16](Esplà-Gomis et al. 2011), and its integration in an Apertium-based interactive machine translation system being developed by Juan Antonio Pérez-Ortiz at the Universitat d'Alacant (this system would provide predictive Kazakh text completions based on the English source text as the professional translators type the translation of that text).

As a longer-range objective, and when a reasonably complete prototype is available, we will tackle another interesting goal: the use of feedback from professional post-editing to improve the system.

[16] http://www.dlsi.ua.es/~mespla/edithints.html

Acknowledgements

We thank Francis M. Tyers, Jonathan North Washington and Ilnar Selimcan for their help during the building of this prototype. The help of KazNU master students Aizhan Aitkulova and Anel Jexekova during the first week of work is also gratefully acknowledged.

References

[1] Cutting, D., Kupiec, J., Pedersen, J., Sibun, P. (1992). A practical part-of-speech tagger. *In*: Proceedings of the 3rd Conference on Applied Natural Language Processing (ANLP'92), p. 133-140.

[2] Esplà-Gomis, M., Sánchez-Martínez, F., Forcada, M.L. (2011). Using machine translation in computer-aided translation to suggest the target-side words to change. In Proceedings of the 13th Machine Translation Summit, p. 172-179, September 19-23, Xiamen, China.

[3] 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.

[4] Ginestí-Rosell, M., Ramírez-Sánchez, G., Ortiz-Rojas, S., Tyers, F. M., Forcada, M. L. (2009). Development of a free Basque to Spanish machine translation system. *Procesamiento del Lenguaje Natural* 43:187-195. URL: http://sepln.org/revistaSEPLN/revista/43/articulos/art21.pdf

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

[6] Lindén, K., Silfverberg, M., Pirinen, T. (2009). Hfst tools for morphology—an efficient open-source package for construction of morphological analyzers. *In*: Mahlow, C., Piotrowski, M. (eds.) State of the Art in Computational Morphology: Workshop on Systems and Frameworks for Computational Morphology, SFCM 2009, Zurich, Switzerland, September 4, 2009. Col. Lecture Notes in Computer Science, 41, p.28—47. Springer.

[7] Sánchez-Martínez, F., Pérez-Ortiz, J. A., Forcada, M. L. (2008). Using target-language information to train part-of-speech taggers for machine translation. *Machine Translation*, 22 (1-2) 29-66.