

A free/open-source Kazakh-Tatar machine translation system

Ilmar Salimzyanov

Affiliation / Address line 1

Affiliation / Address line 2

Affiliation / Address line 3

email@domain

Jonathan North Washington

Departments of Linguistics

and Central Eurasian Studies

Indiana University

Bloomington, Indiana

jonwashi@indiana.edu

Francis Morton Tyers

Affiliation / Address line 1

Affiliation / Address line 2

Affiliation / Address line 3

email@domain

Abstract

This paper presents a bidirectional machine translation system between Kazakh and Tatar.

1 Introduction

This paper presents a prototype shallow-transfer rule-based machine translation system between Kazakh and Tatar.

The paper will be laid out as follows: Section 3 gives a brief description of the two languages; Section 2 gives a short review of some previous work in the area of Turkic-Turkic language translation; Section 4 describes the system and the tools used to construct it; Section 5 gives a preliminary evaluation of the system; and finally Section 6 describes our aims for future work and some concluding remarks.

2 Previous work

Within the Apertium project, work on MT systems between Turkic languages has been started (Turkish-Kyrgyz, Azeri-Turkish), but the Kazakh-Tatar system described by the present study is the closest to production-ready of them. Among these systems is a prototype Tatar-Bashkir machine translation system which was built by the authors of this paper (Tyers et al., 2012); due to the closeness of these languages, it proved to provide high accuracy in its translations, but being a prototype system by design, had relatively low coverage.

Besides these systems, several previous works on making machine translation systems between Turkic languages exist, although to our knowledge none are publicly available. Some MT systems have been reported that translate between Turkish and other Turkic languages, including Turkish-Crimean Tatar (Altintas, 2001b), Turkish-Azerbaijani (Hamzaoglu, 1993), Turkish-Tatar (Gilmullin, 2008), and Turkish-Turkmen

(Tantuğ et al., 2007), though none of these have been released to a public audience.

Within the Apertium project, work on several Turkic-language MT systems has been started, with the ultimate goal of creating independent but compatible finite-state transducers for each language.

3 Languages

Both Tatar and Kazakh belong to the Kypchak (or North-western) group of Turkic languages. They share some level of mutual intelligibility to naïve native speakers, though this is somewhat limited, especially regarding the written languages.

Kazakh is primarily spoken in Kazakhstan, where it is the national language, sharing official status with Russian as the FIXME language. Large groups of native speakers also exist in China, neighbouring Central-Eurasian republics, and Mongolia. The total number of speakers is at least 10 million people.

Tatar is spoken in and around Tatarstan by approximately 6 million people. It is co-official with Russian in Tatarstan – a republic within Russia. A majority of native speakers of both languages are bilingual in Russian.

3.1 Phonological differences

As closely related languages, Kazakh and Tatar share many phonological processes, including front-back vowel harmony systems, consonant voicing assimilation, and even a typologically rare consonantal nasal harmony system. However, the differing details of these processes and the existence of processes unique to each language render Kazakh and Tatar fairly different. For example, Kazakh has a ubiquitous system of desonorisation of the initial sonorants found in many common morphemes. Furthermore, Tatar has nasal assimilation of the initial /l/ of the plural-suffix.

3.2 Orthographic differences

a generalised summary, 1 or 2 small specific examples

3.3 Morphotactic differences

2 or 3 specific examples

3.4 Lexical and morphological differences

2 or 3 specific examples

3.5 Syntactic differences

2 or 3 specific examples > ұна- vs. ярат- (does this count as syntactic?)

4 System

The system is based on the Apertium machine translation platform (Forcada et al., 2011).¹ 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 Licence² (GPL) and all the software and data for the 30 supported language pairs (and the other pairs being worked on) is available for download from the project website.

4.1 Architecture of the system

The Apertium translation engine consists of a Unix-style *pipeline* or *assembly line* with the following modules (see Fig. 1):

- A *deformatter* which encapsulates the format information in the input as *superblanks* that will then be seen as blanks between words by the other modules.
- A *morphological analyser* which segments the text in surface forms (SF) (*words*, or, where detected, multi-word lexical units or MWLUs) and for each, delivers one or more *lexical forms* (LF) consisting of *lemma*, *lexical category* and morphological information.
- A *morphological disambiguator* (constraint grammar) which chooses, using linguistic rules the most adequate sequence of morphological analyses for an ambiguous sentence.
- A *lexical transfer* module which reads each SL LF and delivers the corresponding target-language (TL) LF by looking it up in a bilingual dictionary encoded as an FST compiled from the corresponding XML file. The lexical transfer module may return more than one TL LF for a single SL LF.

- A *lexical selection* module which chooses, based on context rules the most adequate translation of ambiguous source language LFs.
- 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.
- A *morphological generator* which delivers a TL SF for each TL LF, by suitably inflecting it.
- A *reformatter* which de-encapsulates any format information.

4.2 Morphological transducers

The morphological transducers are based on the Helsinki Finite State Toolkit (Linden et al., 2011), a free/open-source reimplement of the Xerox finite-state toolchain, 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. It also supports other finite state transducer formalisms such as **sfst**. This toolkit has been chosen as it – or the equivalent XFST – has been widely used for other Turkic languages (Çöltekin, 2010; Altintas, 2001a; Tantuğ et al., 2006), 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**.

Use of **lexc** allows for straightforward definition of different word classes and subclasses. For example, Tatar (but not Bashkir) has two classes of verbs: one which take a harmonised high vowel in the infinitive (the default), and one which take a harmonised low vowel in the infinitive. This was implemented in **lexc** with two similar continuation lexica for verbs: one pointing at a lexicon with an A-initial infinitive ending, and another pointing at a lexicon with an I-initial infinitive ending.

Use of **twol** allows for phonological processes present in the languages, like vowel harmony and desonorisation, to be implemented in a straightforward manner. For example, in Tatar, the A and I archiphonemes found in the infinitive are harmonised to one of two vowels each, depending on the value of the preceding vowel; the basic form of this process can be implemented in one **twol** rule.

The same morphological description is used for both analysis and generation. To avoid overgeneration, any alternative forms are marked with one of two marks, **LR** (only analyser) or **RL** (only generator). Instead of the

¹<http://www.apertium.org>

²<http://www.fsf.org/licenses/licenses/gpl.html>

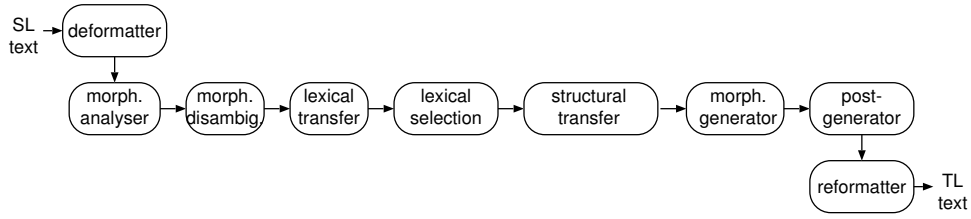


Figure 1: The pipeline architecture of the Apertium system.

usual compile/invert to compile the transducers, we compile twice, once the generator, without the LR paths, and then again the analyser without the RL paths.

4.3 Bilingual lexicon

The bilingual lexicon currently contains 2,834 stem to stem correspondences and was build by hand by a bilingual speaker of Tatar and Bashkir, translating a frequency list of the Russian National Corpus³ into both languages in a spreadsheet. This spreadsheet was then converted into the Apertium XML dictionary format.

Entries consist largely of one-to-one stem-to-stem correspondences with part of speech, but also include some entries with ambiguous translations (see e.g., Fig. 2).

4.4 Disambiguation rules

The system has a morphological disambiguation module in the form of a Constraint Grammar (CG) (Karls-son et al., 1995). The version of the formalism used is vislcg3.⁴

The grammar currently has only four rules, but given the closeness of the languages, the majority of ambiguity may be passed through from one language to the other.

4.5 Lexical selection rules

Likewise, lexical selection is not a large problem between Tatar and Bashkir, but a number of rules can be written for ambiguous words; for example, the Tatar word *борын* ‘nose (person), nose (ship)’ can be translated into Bashkir as either *манау* ‘nose (person)’ or *морон* ‘nose (ship)’. A lexical selection rule chooses the translation *манау* if the immediate context includes a proper name.

Another example is the word *катлаулы* ‘layered’. It is always translated to Bashkir as *катмарлы*, except in the collocation *катлаулы мәсьәлә* ‘difficult matter/problem’, which is translated as *катлаулы мәсьәлә*.

5 Evaluation

Lexical coverage of the system is calculated over a freely available corpus of Bashkir, the Bashkir Wikipedia,⁵ and over two freely available corpora of Tatar, the Tatar Wikipedia⁶ and the New Testament in Tatar. The version of the translation tested was r37137 from the Apertium SVN.⁷

As shown in Table 2, the coverage is still far too low to be of use as a general broad-domain MT system, but we hope that it shows that a good proportion of the morphology of both languages is in place.

To get an idea of the kind of performance that could be expected from the system, we translated a simple story from Tatar to Bashkir and vice versa. The story may be found online,⁸ and was used for pedagogical purposes in a recently workshop on MT for the languages of Russia.

Table 3 presents the Word Error Rate, an edit metric based on the Levenshtein distance (Levenshtein, 1966). This measure was calculated once all the stems in the text had been added to the system, thus presents an upper bound on the current performance of the transfer lexicon, and the disambiguation and transfer rules. The difference in the number of unknown words between translating Tatar→Bashkir and vice versa is because certain forms were not found due to lack of corresponding morphophonological rules.

We calculate the WER instead of other MT evaluation metrics such as BLEU as the WER is geared towards a particular task, that of measuring postedition effort. The translations of the story into Tatar and Bashkir were done in parallel to make them as close as possible, so using BLEU would give an over-optimistic view of the quality.

5.1 Error analysis

The majority of errors are currently due to mistakes and gaps in the morphophonology component; some minor

³<http://ruscorpora.ru/en/>

⁴http://beta.visl.sdu.dk/constraint_grammar.html

⁵<http://ba.wikipedia.org/>; bawiki-20111210-pages-articles.xml.bz2

⁶<http://tt.wikipedia.org/>; ttwiki-20111215-pages-articles.xml.bz2

⁷<https://apertium.svn.sourceforge.net/svnroot/apertium>

⁸<https://apertium.svn.sourceforge.net/svnroot/apertium/branches/xupaixkar/rasskaz>

<e><p><l>юнәләш<s n="n"/></l><r>йүнәләш<s n="n"/></r></p></e>
 <e><p><l>борын<s n="n"/></l><r>танау<s n="n"/></r></p></e>
 <e><p><l>борын<s n="n"/></l><r>морон<s n="n"/></r></p></e>
 <e><p><l>ераклык<s n="n"/></l><r>алыҫлык<s n="n"/></r></p></e>
 <e><p><l>ераклык<s n="n"/></l><r>йыраклык<s n="n"/></r></p></e>

Figure 2: Example entries from the bilingual transfer lexicon. Tatar is on the left, and Bashkir on the right

(Tatar) Input	Һава бүген бик әйбәт, жылы гына.
Mor. analysis	$\hat{h}ава/\hat{h}ава<n><attr>/\hat{h}ава<n><nom>\$ \hat{b}үген/\hat{b}үген<adv>\$$ $\hat{b}ик/\hat{b}ик<adv>/\hat{b}ик<n><attr>/\hat{b}ик<n><nom>\$$ $\hat{a}йбәт/\hat{a}йбәт<adj>/\hat{a}йбәт<adj><subst><nom>\$, /, <cm>\$$ $\hat{j}ылы/\hat{j}ылы<n><attr>/\hat{j}ылы<n><nom>/\hat{j}ылы<adj>/\hat{j}ылы<adj><subst><nom>\$$ $\hat{g}ына/\hat{g}ына<postadv>\$, /, <sent>\$$
Mor. disambiguation	$\hat{h}ава<n><nom>\$ \hat{b}үген<adv>\$ \hat{b}ик<adv>\$ \hat{a}йбәт<adj>\$, /, <cm>\$$ $\hat{j}ылы<adj>\$ \hat{g}ына<postadv>\$, /, <sent>\$$
Lex. transfer (+ selection)	$\hat{h}ава<n><nom>/\hat{h}ауа<n><nom>\$ \hat{b}үген<adv>/\hat{b}өгөн<adv>\$ \hat{b}ик<adv>/\hat{b}ик<adv>\$$ $\hat{a}йбәт<adj>/\hat{a}йбәт<adj>\$, /, <cm>\$, /, <cm>\$ \hat{j}ылы<adj>/\hat{j}ылы<adj>\$$ $\hat{g}ына<postadv>/\hat{g}ына<postadv>\$, /, <sent>\$, /, <sent>\$$
Struct. transfer	$\hat{h}ауа<n><nom>\$ \hat{b}өгөн<adv>\$ \hat{b}ик<adv>\$ \hat{a}йбәт<adj>\$, /, <cm>\$$ $\hat{j}ылы<adj>\$ \hat{g}ына<postadv>\$, /, <sent>\$$
Mor. generation	Һауа бөгөн бик әйбәт, йылы гына.

Table 1: Translation process for the phrase *Һава бүген бик әйбәт, жылы гына* ‘The weather today is very nice, it is very warm’.

problems still remain involving:

- Combinations of case and possessive suffixes,
- Orthographical representations of phonology,
- Vowel harmony processing on clitics (e.g., *да/да* ‘and’) after unknown words.

6 Concluding remarks

To our knowledge we have presented the first ever MT system between Tatar and Bashkir, and the first ever MT system involving Bashkir. The system is available as free/open-source software under the GNU GPL and the whole system may be downloaded from SVN.⁹

We plan to continue development on the pair; the main work will be expanding the dictionaries with new lists of stems, and providing bilingual correspondences. The long-term plan is to integrate the data created with other open-source data for Turkic languages in order to make transfer systems between all the Turkic language pairs. Related work is currently ongoing with Chuvash–Turkish and Turkish–Kyrgyz.

⁹<https://apertium.svn.sourceforge.net/svnroot/apertium/nursery/apertium-tt-ba>

Corpus	Tokens	Coverage
Tatar New Test.	163,603	72.04%
Tatar Wikipedia	37,123	70.19%
Bashkir Wikipedia	12,267	65.99%

Table 2: Naïve vocabulary coverage over the three corpora.

Corpus	Direction	Tokens	Unknown	WER
story	tt→ba	311	9	8.97%
	ba→tt	312	1	7.72%

Table 3: Word error rate and over the small test corpus.

Acknowledgements

We would like to thank the anonymous reviewers for their helpful comments in improving the paper. This work has been partially funded by Spanish Ministerio de Ciencia e Innovación through project TIN2009-14009-C02-01.

References

- Altintas, K. (2001a). A Morphological Analyser for Crimean Tatar. *Proceedings of Turkish Artificial Intelligence and Neural Network Conference*.
- Altintas, K. (2001b). Turkish To Crimean Tatar Machine Translation System. Master's thesis, Bilkent University.
- 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. A. (2008). The Tatar-Turkish Machine Translation Based On The Two-Level Morphological Analyzer. In *Interactive Systems and Technologies : The Problems of Human-Computer Interaction*, pages 179–186, Ulyanovsk.
- Hamzaoglu, I. (1993). Machine translation from Turkish to other Turkic languages and an implementation for the Azeri language. Master's thesis, Bogazici University.
- Karlsson, F., Voutilainen, A., Heikkilä, J., and Anttila, A. (1995). *Constraint Grammar: A language independent system for parsing unrestricted text*. Mouton de Gruyter.
- Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics—Doklady* 10, 707–710. Translated from *Doklady Akademii Nauk SSSR*, pages 845–848.
- Linden, K., Silfverberg, M., Axelson, E., Hardwick, S., and Pirinen, T. (2011). *HFST—Framework for Compiling and Applying Morphologies*, volume Vol. 100 of *Communications in Computer and Information Science*, pages 67–85.
- Tantuğ, A. C., Adalı, E., and Oflazer, K. (2007). A MT system from Turkmen to Turkish employing finite state and Statistical Methods. In *Proceedings of MT Summit XI, Copenhagen, Denmark*.
- Tantuğ, A. C., Adalı, E., and Oflazer, K. (2006). Computer Analysis of Turkmen Language Morphology. pages 186–193. *Advances in natural language processing, proceedings (LNAI)*.
- Tyers, F., Washington, J. N., Salimzyan, I., and Batalov, R. (2012). A prototype machine translation system for Tatar and Bashkir based on free/open-source components. In *Proceedings of the First Workshop on Language Resources and Technologies for Turkic Languages at the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey.
- Çöltekin, İ. (2010). A freely available morphological analyzer for Turkish. *Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC2010)*, pages 820–827.