Multi-script morphological transducers and transcribers for seven Turkic languages

Jonathan Washington, Francis Tyers, Oğuzhan Kuyrukçu
Swarthmore College, Indiana University / Высшая Школа Экономики, Boğaziçi Üniversitesi
jonathan.washington@swarthmore.edu, ftyers@iu.edu, kuyrukcuoguz@gmail.com

This paper describes ongoing work to augment morphological transducers for seven Turkic languages with support for multiple scripts each, as well as respective IPA transcription systems. Evaluation demonstrates that our approach yields coverage equivalent to or not much lower than that of the base transducers.

Background. A morphological transducer converts between form and analysis, e.g. `алмалардан` ↔ `алма<n><pl><abl>`, where the form-to-analysis task is termed "morphological analysis" and the analysis-to-form task is termed "morphological generation". Existing Free/Open-Source morphological transducers for Turkic languages (Washington et al. 2020) are implemented in only one orthography, despite a number of the languages being currently written in two or more orthographies, or having a large body of text written in an orthography that was recently switched away from.

This paper builds on work in which Cyrillic support was added to a transducer for Crimean Tatar which had been implemented in the Latin script (Tyers et al. 2019). We leverage morphological transducers for Kazakh (implemented in the Cyrillic script), Kyrgyz (Cyrillic), Turkmen (Latin), Qaraqalpaq (Latin), Uzbek (Latin), and Uyghur (Perso-Arabic), and add support for analysis and generation in additional scripts that are currently or have recently been used for the languages. Specifically, we add Cyrillic support to Turkmen, Qaraqalpaq, Uzbek, and Uyghur transducers; Perso-Arabic support to the Kazakh and Kyrgyz transducers; Latin script to the Kazakh and Uyghur transducers; and IPA support to all of them.

Motivation for this work is plentiful. Support for multiple scripts allows academic work that employs corpora of these languages to expand to include text from more sources. More importantly, language technology that uses these transducers as a component, such as machine translation systems and spell checkers, will now be accessible to more communities that use the languages. Additionally, these systems may be used to simply convert between the orthographies, a task that has the potential to enable textual exchange between different communities using a single language with different scripts. We further expect the transcribers to be useful in the development of text-to-speech systems for these languages.

Methodology. Our approach employs HFST (Linden et al. 2011) to create a transliteration transducer, consisting of two transducers compose-intersected: (1) a lexical transducer which maps all possible combinations of all characters used in the language, and (2) a two-level phonology finite-state automaton, where the "phonology" consists of character mappings sensitive to context. This transliteration transducer is then compose-intersected with the existing transducer to create a transducer in the new script; the resulting transducer is then unioned with the original transducer to create a transducer that can analyse both scripts. The steps are repeated for a third script to create tri-scriptual transducers, and so on.

This approach presents simple ways of dealing with common problems in mapping between orthographies. For example, many of the Cyrillic orthographies used have letters that represent the glide /j/ followed by a vowel, whereas the Latin and Perso-Arabic orthographies write /j/ and a vowel as two separate characters. One of the easier ways to solve this is to map each /jV/ Cyrillic letter to two Cyrillic letters in the lexical transducer, e.g. `я:йа`.

Another potential challenge is including a *dayekshe* at the beginning of front-vowel words in Kazakh in which an unambiguously "front" letter is not present. This is solved by inserting a special symbol at the beginning of all words in the lexical transducer, deleting it by default in the phonological transducer, and restricting it to output as a *dayekshe* only in words where a Cyrillic front-vowel letter that is ambiguous in backness in the Perso-Arabic script is used and no unambiguously front character is in the word. Another common problem is that some characters are represented multiple ways depending on where the text is taken from. For example, the apostrophe-like diacritic in Qaraqalpaq may be represented as an apostrophe (U+0027), a turned comma (U+02BB), modifier letter apostrophe (U+02BC),

among others. Another example is the Kazakh *dayekshe*, which is alternatingly represented as U+0621 (hamza) or U+0674 (high hamza). For situations like these, we implement an additional two-level phonology transducer which is compose-intersected with the analyser transducer so that multiple variants may be recognised simultaneously.

One problem encountered in transcription transducers is the inclusion of stress marks in the correct place. Since stress in many Turkic languages may be morphologically and lexically conditioned, the obvious solution includes adding stress information to the base transducers that is normally hidden.

A more complicated challenge is when there are multiple possibilities for converting a single character to another script. For example, the Kyrgyz Cyrillic script, in which the original transducer is designed, represents both /q/ and /k/ with the same character (к), but the Kyrgyz Perso-Arabic script uses two separate characters (ك and ق). In this case, the choice can usually be conditioned on surrounding vowels.

A somewhat more complicated version of this problem is generating Russian words correctly in Russian orthography from Crimean Tatar Latin orthography, which does not have equivalents for characters such as hard sign and soft sign—e.g., ‹yanvar› 'January' as ‹январь› instead of *‹январ›. To increase the level of determinism in the conversion, an *n*-gram language model is used to weight the Cyrillic side of the end transducer, essentially biasing it towards Russian spellings.

A variant of this problem is correctly generating forms with correct letter casing when the original transducer is implemented in the Perso-Arabic script (which does not have case), as has arisen with the Uyghur transducer. We are still working on a solution to this challenge, but it will likely involve an *n*-gram language model, similarly to how the non-determinism in Latin-to-Cyrillic conversion for Crimean Tatar has been dealt with.

Evaluation. We report two forms of evaluation for these multi-scriptual transducers: naïve coverage (the number of forms in a corpus for which an analysis is returned, whether correct or not) over equivalent corpora for multiple scripts, and an in-depth analysis of the reasons for which forms in the corpus did not receive analyses.

The main reasons forms were not analysed were (1) that they were not present in the base transducer or (2) there were errors in the orthographic conversion. The reasons that forms are not present in the base transducer may be (1a) because the forms are not used in regions where the base transducer's script is used, (1b) because of idiosyncratic differences in orthographic or phonological treatment of equivalent forms, (1c) because the forms were simply missing from the base transducer, or (1d) because the corpus contains forms that are not correct forms in the language (such as document markup symbols and content in other languages).

An example of (1a) is that Kazakh texts in the Perso-Arabic script (as used in China) may contain loanwords from Chinese that Kazakh texts in Cyrillic (the orthography of the base transducer, as used in Kazakhstan) do not.

Because only reasons (1c) and (1d) affect the base transducer, alternate-orthography transducers created using our approach normally support at most the forms in the base transducer, and as a result have at best equivalent coverage numbers. While errors in orthographic conversion (2) may hypothetically yield [incorrect] analyses for forms in the additional orthography that would not receive an analysis in the base transducer's orthography, this is extremely rare.

Future work will continue to address problems in all of these areas.

References

Linden, Krister, Miikka Silfverberg, Erik Axelson, Sam Hardwick, and Tommi Pirinen (2011). "HFST—Framework for Compiling and Applying Morphologies". In: *Systems and Frameworks for Computational Morphology*. Ed. by Cerstin Mahlow and Michael Pietrowski. Vol. 100. Communications in Computer and Information Science, pp. 67–85.

Tyers, Francis M., Jonathan Washington, Darya Kavitskaya, Memduh Gökırmak, Nick Howell, and Remziye Berberova (2019). "A biscriptual morphological transducer for Crimean Tatar". In: vol. 1, pp. 74–80. URL: https://scholar.colorado.edu/scil-cmel/vol1/iss1/10.

Washington, Jonathan N., Ilnar Salimzianov, Francis M. Tyers, Memduh Gökırmak, Sardana Ivanova, and Oğuzhan Kuyrukçu (2020). "Free/Open-Source technologies for Turkic languages developed in the Apertium project". In: *Proceedings of the Seventh International Conference on Turkic Language Processing (TurkLang 2017)*. in press.