A biscriptual morphological transducer for Crimean Tatar

Francis M. Tyers
Department of Linguistics
Indiana University
ftyers@iu.edu

Jonathan North Washington
Linguistics Department
Swarthmore College
jonathan.washington@swarthmore.edu

Darya Kavitskaya
Slavic Languages and Literatures
U.C. Berkeley
dkavitskaya@berkeley.edu

Memduh Gökırmak
ÚFAL
Univerzita Karlova in Prague
memduhg@gmail.com

Nick Howell
Faculty of Mathematics
Higher School of Economics
nlhowell@gmail.com

Remziye Berberova
Department of Crimean Tatar*
Crimean Tavrida University
bra.berberova@yandex.ua

Abstract

This paper describes a weighted finite-state morphological transducer for Crimean Tatar able to analyse and generate in both Latin and Cyrillic orthographies. This transducer was developed by a team including a community member and language expert, a field linguist who works with the community, a Turkologist with computational linguistics expertise, and an experienced computational linguist with Turkic expertise.

Dealing with two orthographic systems in the same transducer is challenging as they employ different strategies to deal with the spelling of loan words and encode the full range of the language's phonemes and their interaction. We develop the core transducer using the Latin orthography and then design a separate transliteration transducer to map the surface forms to Cyrillic. To help control the non-determinism in the orthographic mapping, we use weights to prioritise forms seen in the corpus. We perform an evaluation of all components of the system, finding an accuracy above 90% for morphological analysis and near 90% for orthographic conversion. This comprises the state of the art for Crimean Tatar morphological modelling, and, to our knowledge, is the first biscriptual single morphological transducer for any language.

1 Introduction

This paper presents the development and evaluation of a free/open-source finite-state morphological transducer for Crimean Tatar that is able to analyse and generate both Latin and Cyrillic orthographies.[1] As an example, this transducer can map between the analysis `köy<n><px3sp><loc>` 'in the village of' and both of the possible orthographic forms, in Latin *köyünde* and in Cyrillic *коюнде*.

This paper is structured as follows: Section 2 presents background on Crimean Tatar and its orthographic systems; Section 3 gives an overview of the current state of computational analysis of Crimean Tatar morphology; Section 4 presents certain challenging aspects of Crimean Tatar morphology and their implementation; and Section 5 evaluates the transducer. Finally, Section 6 summarises possible directions for future work, and Section 7 provides concluding remarks.

2 Crimean Tatar

2.1 Context

Crimean Tatar ([qɯɾɯmtɑtɑɾʧɑ], ISO 639-3: `crh`) is an understudied Turkic language of the Northwestern Turkic (Kypchak) subgroup (Johanson and Csató, 1998).[2] The language also shows a considerable influence from the Southwestern Turkic (Oghuz) subgroup, acquired via contact with Turkish, as well as more recent influence from the Southeastern Turkic subgroup, due to the nearly 5-decade-long resettlement of the entire Crimean Tatar population of Crimea to Central Asia (predominantly to Uzbekistan) by the Soviet government in 1944. It shares some level of mutual intelligibility with other languages in these subgroups, but is an independent language. The geographical location of Crimean Tatar in reference to other NW (Kypchak) and SW (Oghuz) varieties is shown in Figure 1.

Currently, about 228,000 speakers of Crimean Tatar have returned to Crimea, and another 313,000 live in diaspora (Simons and Fennig, 2018). Almost all speakers of Crimean Tatar are bilingual or multilingual in Russian and the language of the place of their exile, such as Uzbek, Kazakh, or Tajik.

†Emerita

[1] The transducer is available in a publicly accessible repository at http://github.com/apertium/apertium-crh.

[2] Crimean Tatar and [Kazan] Tatar (`tat`) happen to share a name, but are only related in that they are both Northwestern Turkic languages—though members of different subsubgroups.

Proceedings of the 3rd Workshop on the Use of Computational Methods in the Study of Endangered Languages: Vol. 1 Papers, pages 74–80,
Honolulu, Hawai'i, USA, February 26–27, 2019.

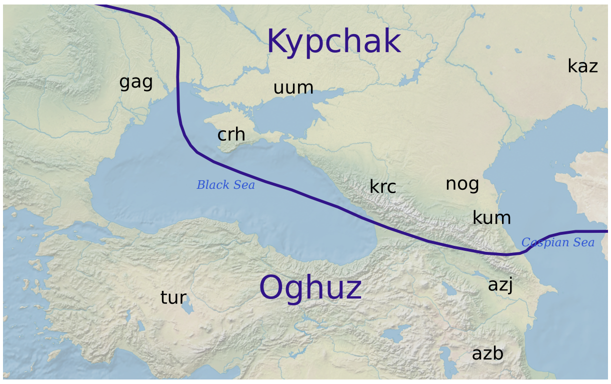

Figure 1: Location of the Crimean Tatar speaking area (crh) within the Black Sea region, relative to other Kypchak (Urum – uum, Karachay-Balkar – krc, Nogay – nog, Kumyk – kum and Kazakh – kaz) and Oghuz languages (Gagauz – gag, Turkish – tur, and Azerbaijani – azb and azj).

Crimean Tatar is spoken mostly by the older population, but the usage may be rising, due to the increasing efforts in the community to teach the language to the younger generation, despite impedance caused by the generally unfavourable sociolinguistic situation.

2.2 Orthographic systems

As previously mentioned, Crimean Tatar can be written with two orthographic systems, one based on the Latin alphabet and one based on the Cyrillic alphabet. Both orthographies are widely used and have varying degrees of official support. They have different ways of treating phenomena such as front rounded vowels, the uvular/velar contrast in obstruents, and loanwords from Russian.

The Latin orthography contains 31 characters, and uses no digraphs except for foreign sounds in borrowings. Each phoneme is implemented using a distinct character; for example ‹o› for the non-high back rounded vowel and ‹ö› for its front counterpart. The diacritics tilde (‹ñ› for [ŋ]), cedilla (e.g., ‹ş› for [ʃ]) and diaeresis (e.g., ‹ö› for [ø]) are used as components of characters for sounds that are not covered in a straightforward way by the basic Latin alphabet. In the Latin orthography, Russian words can be treated as adapted to the phonology of Crimean Tatar, such as *bücet* ‘budget’—as opposed to **byudjet*, a more faithful rendering in the Latin orthography of the pronunciation of the Russian *бюджет*. However, most loanwords are pronounced as in Russian, yet are rendered more faithfully to their Russian spelling than to their Russian pronunciation; for example, *konki* for Russian *коньки* [kɐnʲˈkʲi] ‘skates’ is pronounced as in Russian and not *[konki] as its Latin-orthography spelling would suggest.

The Cyrillic orthography contains the 33 characters of the Russian alphabet and four diagraphs for sounds not found in Russian: ‹дж› [ʤ], ‹гъ› [ʁ], ‹къ› [q], and ‹нъ› [ŋ]. The additional sounds [ø] and [y] are implemented by either the use of the “soft sign” ‹ь› in conjunction with the letters for [o] ‹o› and [u] ‹y› or by using the corresponding “yoticised” vowel letters ‹ë› and ‹ю›, respectively; the particular system is clarified below. Russian words are spelled as in Russian, including the use of hard and soft signs and Russian phonological patterns. An example is *коньки* ‘skates’, in which the ‹ь› indicates that the [n] is palatalised.

Table 1 shows the basic mapping between the two orthographic systems.

b	c	ç	d	f	g	ğ	h	j	k	l
б	дж	ч	д	ф	г	гъ	х	ж	к	л

m	n	ñ	p	q	r	s	t	v	y	z
м	н	нъ	п	къ	р	с	т	в	й	з

a	â	ı	o	u	e	i	ö	ü
а, я	я	ы	о, ё	у, ю	э, е	и	о, ё	у, ю

Table 1: The basic correspondences between the characters of the two Crimean Tatar orthographies.

While the Latin orthography represents the front vowels [ø] and [y] simply as ‹ö› and ‹ü›, in the Cyrillic orthography they are represented in one of the following ways.

With the letters ‹o› and ‹y›:

- With a front-vowel character (‹и›, ‹e›) or one of the “yoticised” vowels (‹ë›, ‹ю›) following the subsequent consonant. This strategy is usually limited to the first syllable of a [multisyllable] word when in certain consonant contexts, and is more prevalent in open syllables. Examples include *учюнджи* [yʧ-ynʤi] ‘third’, *кунюнде* [kyn-yn-de] ‘on the day of’, *болип* [bøl-ip] ‘having divided’, and *муче* [myʧe] ‘body part’.

- Either with the “soft sign” ‹ь› following the subsequent consonant or without if a “soft consonant” (i.e., velar pair to a uvular consonant), like [k] or [g], follows the vowel. This strategy is generally used only in the first syllable of a word when there is a following coda consonant. Examples include учь [yʧ] ‘three’, *ку̲ньлери* [kyn-ler-i] ‘days of’, *бо̲льмеди* [bøl-me-di] ‘did not divide’, *ко̲к* [køk] ‘sky’, and *бу̲кти* [byk-ti] ‘he/she/it bent’.

With the “yoticised” vowel letters ‹ё› and ‹ю›:

- When not in the first syllable of a word. Examples include *юзю̲* [jyz-y] ‘his/her face’, *тёксю̲н* [tøk-syn] ‘let it spill’, and *мумкю̲н* [mymkyn] ‘possible’. Note that [ø] almost never occurs outside of the first syllable of a word.

- When in the first syllable of a word and preceded by a consonant. Examples include *тю̲шти* [tyʃ-ti] ‘fell’, *дё̲рт* [dørt] ‘four’, and *чё̲пке* [ʧøp-ke] ‘to the rubbish’.

The difference between when the letters ‹о› or ‹у› are used as opposed to ‹ё› or ‹ю› in first syllables of words seems to in part depend on the values of surrounding consonants. However, a certain level of idiosyncrasy exists, as seen in pairs like *козь* [køz] ‘eye’ and *сёз* [søz] ‘word’, *кёр* [kør] ‘blind’ and *корь* [kør] ‘see’, or *юз* [jyz] ‘hundred’ and *юзь* [jyz] ‘face’.

The “yoticised” vowel letters ‹ё› and ‹ю› also represent the vowels [o] and [u] when following [j] as well as [ø] and [y] when following [j]—sound combinations that can occur word-initially or after another vowel (usually rounded and of corresponding backness, though note that [ø] and [o] are extremely uncommon outside of the first syllable of a word). Hence there is in principle the potential for systematic ambiguity between rounded back vowels preceded by [j] (‹ё› [jo] and ‹ю› [ju]) and rounded front vowels preceded by [j] (‹ё› [ø] and ‹ю› [y]). In practice it is difficult to identify examples of this, but pairs like *ют* [jut] ‘swallow’ and *юз* [jyz] ‘hundred’ demonstrate the concept.

Furthermore, [j] is represented in the Cyrillic orthography either by й (e.g., *къой̲* [qoj] [put]), or with a yoticised vowel letter (e.g., *къою̲л* [qoj-ul] ‘be put’); i.e., these letters are involved in a many-to-many mapping with the phonology.

3 Prior work

Altıntaş and Çiçekli (2001) present a finite-state morphological analyser for Crimean Tatar. Their morphological analyser has a total of 5,200 stems and the morphotactics[3] are based on a morphological analyser of Turkish. They explicitly cover only the native part of the vocabulary, excluding loan words, and use an ASCII representation for the orthography. Their analyser is not freely available for testing so unfortunately we could not compare its performance to that of ours.

4 Methodology

To implement the transducer we used the Helsinki Finite-State Toolkit, HFST (Lindén et al., 2011). This toolkit implements the `lexc` and `twol` formalisms and also natively supports weighted FSTs. The former implements morphology, or mappings between analysis and morphological form, such as `köy<n><px3sp><loc>` : `köy>{s}{I}{n}>{D}{A}`, while the latter is used to ensure the correct mapping between morphological form and orthographic (or “phonological”) form, such as `köy>{s}{I}{n}>{D}{A}` : `köyünde`. When compose-intersected, the transducers generated from these modules result in a single transducer mapping the two ends with no intermediate form, e.g., `köy<n><px3sp><loc>` : `köyünde`.

The choice to model the morphophonology using `twol` as opposed to using a formalism that implements sequential rewrite rules may be seen as controversial. Two-level phonological rules are equivalent in expressive power to sequential rewrite rules (Karttunen, 1993); however, from the point of view of linguistics, they present some differences in terms of how phonology is conceptualised. Two-level rules are viewed as constraints over a set of all possible surface forms generated by expanding the underlying forms using the alphabet, and operate in parallel. Sequential rewrite rules, on the other hand, are viewed as a sequence of operations for converting an underlying form to a surface form. As such, sequential rules result in intermediate forms, whereas the only levels of representation relevant to two-level rules are the morphological (underlying) form and the phonological (surface) form. While it may not be relevant from an engineering point of view, we find more cognitive plausibility in the two-level approach. Furthermore, the computational phonologist on our team finds the two-level model much less cumbersome to work with for modelling an entire language’s phonology than sequential rewrite rules. Readers are encouraged to review Karttunen (1993) for a more thorough comparison of the techniques.

[3] The morphotactics of a language is the way in which morphemes can be combined to create words.

4.1 Lexicon

The lexicon was compiled semi-automatically. We added words to the lexicon by frequency, based on frequency lists from Crimean Tatar Bible[4] and Wikipedia[5] corpora (see Section 5.1 for sizes). Table 2 gives the number of lexical items for each of the major parts of speech. The proper noun lexicon includes a list of toponyms extracted from the Crimean Tatar Wikipedia.

Part of speech	Number of stems
Noun	6,271
Proper noun	4,123
Adjective	1,438
Verb	1,007
Adverb	87
Numeral	40
Pronoun	31
Postposition	21
Conjunction	20
Determiner	16
Total:	13,054

Table 2: Number of stems in each of the main categories.

4.2 Tagsets

The native tagset of the analyser is based on the tagsets used by other Turkic-language transducers[6] in the Apertium platform.[7] In addition we provide a mapping from this tagset to one compatible with Universal Dependencies (Nivre et al., 2016) based on 125 rules and a set overlap algorithm.[8] The rules are composed of triples of (lemma, part of speech, features) and are applied deterministically longest-overlap first over the source language analyses.

4.3 Morphotactics

The morphotactics of the transducer are adapted from those of the Kazakh transducer described by Washington et al. (2014). The nominal morphotactics are almost identical between Kazakh and Crimean Tatar. The verbal morphotactics are rather different, and we here followed Kavitskaya (2010).

4.4 Transliterator

The transliterator is implemented as a separate substring-to-substring `lexc` grammar and `twol` ruleset.

The `lexc` grammar defines a transducer which converts from the Latin orthography to a string where placeholders are given for the hard sign, soft sign, and some digraphs which are single characters in the Cyrillic orthography (e.g., *ts* = *ц*). The output may be ambiguous, for example the input string *şç* produces both `şç` (analysis, preceding transliteration) and a special symbol щ (also analysis) standing for Cyrillic *щ* (surface form). This is necessary because *şç* may map to either *щ* (*borşç* = *борщ* [borɕ] 'borsch') or ‹шч› (*işçi* = *ишчи* [iʃʧi] 'worker').

The `twol` ruleset defines a transducer which then maps the Latin string produced to strings in Cyrillic via the alphabet, and applies a set of constraints to restrict the possible combinations. All remaining theoretically valid mappings are kept. An example of one of these constraints is shown in Figure 2.

```
"e as э"
e:э <=> .#. _ ;
        [ e: | a: | i: | ü: ] _ ;
```

Figure 2: An example of a `twol` constraint used in the mapping of Latin strings to Cyrillic strings. This constraint forces Latin ‹e› [e] to be realised as Cyrillic ‹э› instead of Cyrillic ‹е› (which would in turn stand for [je]) at the beginning of the word and after certain vowels.

The resulting transducer takes surface forms in Latin script, and outputs surface forms in Cyrillic script. In order to get an analyser which analyses Cyrillic, we then composed the original [Latin] transducer with the transliteration transducer.

In order to be able to choose the orthographically correct variant for generation in the case of ambiguity in the conversion, we tried two corpus-based methods.

The first method we tried was simply to weight surface forms we saw in the corpus with a negative weight; since our transducer interprets lower weights as better, forms which were previously seen would always be given preference over those generated by the model *on the fly*. This was done by making a negative-weight identity transducer of the surface forms, composing with the transliteration transducer, and taking the union with the transliteration transducer alone.

The second method was to estimate probabilities for the generated transliterations using character n-

[4] Compiled by IBT Russia/CIS, `https://ibt.org.ru`

[5] Content dump from the Crimean Tatar Wikipedia, `https://crh.wikipedia.org`, dated `2018-12-01`.

[6] See for example Washington et al. (2016) for a description.

[7] Available online at `http://www.apertium.org`.

[8] Available in the repository as `texts/crh-feats.tsv`.

gram frequencies from the corpus. We used a particularly simplistic estimation technique: fix $n \geq 1$. We collect k-grams for $k \leq n$, the collections of size n_k with redundancy. The probability assigned to the k-gram x is $\#x/n_k$. Weights are chosen to be the negative logarithm of the probability. The unaugmented transliteration transducer is then composed with this new weighted transducer before being composed with the morphological transducer.

Transducers produced by the two methods are unioned; the result is then composed with the morphological transducer. Generated transliterations thus have many paths which they can follow to acceptance, assigning various weights. If the transliteration is a surface form observed in the training corpus, it is assigned a negative weight (given absolute preference). Unobserved transliterations are assigned positive weights based on possible segmentations; k-character segments are assigned weights from the k-gram counters, and the weight of a segmentation is the sum of the weights of the segments.

4.5 Error model

While working on morphological models for languages with a less-than-stable written norm, or where there is little support for proofing tools or keyboards, it is desirable to be generous with what forms are accepted while being conservative with what forms are generated. Orthographic variation is inevitable, and if we want to create a high coverage resource, then we should also take this variation into account. For example, in the corpus the locative of *Belarus* (the country) is written *Belarusde* 4 times, *Belarusta* twice and *Belaruste* 1 time. The normative spelling, to fit with the pronunciation [belarus] should be *Belarusta*; however, we would also like to be able to provide an analysis for the other variants. Based on an informal examination of $1,000$ random tokens in the news and encyclopaedic corpora, we estimate that at least 0.8% of tokens in these corpora constitute non-normative orthographic forms of this type.

Our approach again is to use `twol`. First the rules for vowel harmony and other phonological phenomena were removed from the `twol` transducer that implements the normative orthography, leaving only unconstrained symbol mappings. This was then composed with the lexicon to produce a transducer which has all of the possible phonological variants (much like a fully expanded version of the `lexc` transducer). This was then subtracted from the normative transducer and a tag `<err_orth>` was appended to the analysis side of all remaining forms to indicate orthographic error, This was output into a transducer which accepts any phonological variant that was not normative and give an analysis with an extra tag. This was unioned with the normative transducer to produce the final analyser. This approach allows us to analyse prescriptively incorrect variants like *Belaruste* as `Belarus<np><top><loc><err_orth>`.

5 Evaluation

We have evaluated the morphological transducer in several ways. We computed the naïve coverage and the mean ambiguity of the analyser on freely available corpora (Section 5.1) as well as its accuracy (precision and recall) against a gold standard (Section 5.2). Additionally, we evaluated the accuracy of the transliteration transducer (Section 5.3).

5.1 Analyser coverage

We determined the naïve coverage and mean ambiguity of the morphological analyser. Naïve coverage is the percentage of surface forms in a given corpus that receive at least one morphological analysis. Forms counted by this measure may have other analyses which are not delivered by the transducer. The mean ambiguity measure was calculated as the average number of analyses returned per token in the corpus. These measures for three corpora, spanning both orthographies, are presented in Table 3.[9]

Corp.	Orthog.	Tokens	Cov.	Ambig.
Bible	Cyr	217,611	90.9%	1.86
Wiki	Lat	214,099	92.1%	1.86
News	Lat	1,713,201	93.7%	2.12

Table 3: Naïve coverage and mean ambiguity of the analyser on three corpora.

The transducer provides analyses for over 90% of tokens in each corpus, with each token receiving an average of around two analyses.

5.2 Transducer accuracy

Precision and recall are measures of the average accuracy of analyses provided by a morphological transducer. Precision represents the number of the

[9] The Bible and Wikipedia corpora are those described in Section 4.1, and the News corpus is content for the years 2014–2015 extracted from `http://ktat.krymr.com/`.

analyses provided by the transducer for a form that are correct. Recall is the percentage of analyses that are deemed correct for a form that are provided by the transducer.

To calculate precision and recall, it was necessary to create a hand-verified list of surface forms and their analyses. We extracted around 2,000 unique surface forms at random from the Wikipedia corpus, and checked that they were valid words in the languages and correctly spelled. When a word was incorrectly spelled or deemed not to be a form used in the language, it was discarded.[10]

This list of surface forms was then analysed with the most recent version of the analyser, and around 500 of these analyses were manually checked. Where an analysis was erroneous, it was removed; where an analysis was missing, it was added. This process gave us a 'gold standard' morphologically analysed word list of 448 forms.[11]

We then took the same list of surface forms and ran them through the morphological analyser once more. Precision was calculated as the number of analyses which were found in both the output from the morphological analyser and the gold standard, divided by the total number of analyses output by the morphological analyser. Recall was calculated as the total number of analyses found in both the output from the morphological analyser and the gold standard, divided by the number of analyses found in the morphological analyser plus the number of analyses found in the gold standard but not in the morphological analyser.

After comparing with the gold-standard in this way, precision was 94.98% and recall was 81.32%.

Most of the issues with recall were due to missing stems in the lexicon, primarily nouns and proper nouns.[12] Regarding the precision, common issues included incorrect categorisation in the lexicon, and dubious forms, such as the imperative of *kerek-* 'need', which is in the analyser and is a hypothetically possible form, but appears not to be possible in practice.

5.3 Transliterator accuracy

We evaluated the transliteration component on the headwords from a bilingual Crimean Tatar–Russian dictionary that has been published in both Cyrillic and Latin orthographies.[13] We created a list of Cyrillic–Latin correspondences by aligning the headwords automatically based on an automated word-by-word comparison of the definitions in Russian, for a total of $14,905$ unique entries.

141 entries (~1%) had comments which did not match word-for-word; while it is possible that these could be corrected by hand, we discarded them. We then fed the Latin entries to the full transliteration transducer and evaluated against the corresponding Cyrillic entry.

Table 4 shows the performance of the transliterator for the different methods. In this case, precision is the percentage of predictions which are correct, recall is the percentage of words for which a correct transliteration is predicted, and the F-score is the harmonic mean of the two: $F^{-1} = \text{mean}(\text{Prec}^{-1}, \text{Rec}^{-1})$.

Method	States	Precision	Recall	F-score
–	114	53.0	98.4	68.9
1-gram	2,030	93.4	93.5	93.5
2-gram	17,382	94.1	94.2	94.1
3-gram	99,761	94.0	94.1	94.1
4-gram	290,201	94.4	94.6	94.5
5-gram	577,926	95.1	95.2	95.2
6-gram	924,719	95.5	95.6	95.5
7-gram	1,282,917	95.4	95.6	95.0

Table 4: Performance of the transliterator using different methods. States gives a measure of the size of the generated FST.

Without n-grams, there is no attempt to filter proposed transliterations; that is, this "null" method generates all possible transliterations according to the combined phonological-morphological transducer. It demonstrates the theoretical limit of recall. Precision dramatically increases with the introduction of n-grams, as expected. Precision increases with more n-grams, levelling off at just over 95%. Recall drops from the maximum of 98.4% (the theoretical maximum the n-gram system can hope to attain); as the quality of the statistical filter increases, so does recall, until it levels off at 95.6%.

The problems with the transliteration model consist almost entirely of issues related to the presence of hard and soft signs in Cyrillic spellings (accounting for 492 of 1007, or 48.9%, of errors), incorrect vowels, mostly related to yoticisation (accounting for 469, or 46.6%, of errors), and issues correctly

[10] Available in the repository as `dev/annotate.txt`.

[11] Available in the repository as `dev/annotate.all.txt`.

[12] However, note that the recall number may be somewhat inflated, as thinking of missing analyses for already analysed words is particularly difficult.

[13] Available from `http://medeniye.org/node/984`.

predicting “ц” versus “тс” (accounting for 40, or 4% of errors). These errors typically arise in loanwords, where the correct Cyrillic spelling is often impossible to predict from the Latin orthography. Accuracy regarding these issues could likely be improved by having a larger and more representative corpus of Crimean Tatar in Cyrillic with which to train the n-gram models, or by attempting to model the loanword system.

6 Future work

The performance of the n-gram model could be improved by modelling the predictability of the orthography in n-grams and with a sliding window to filter out unlikely concatenations of common n-grams.

Aside from expanding the lexicon, the transducer forms part of a machine translation system from Crimean Tatar to Turkish being developed in the Apertium platform. There is also the prospect of applying it to dependency parsing for Crimean Tatar, and there have been some preliminary experiments in this direction (Ageeva and Tyers, 2016). We would also like to apply the approach for dealing with multiple scripts to other Turkic languages, such as Uzbek, Kazakh, or Karakalpak, where more than one widely-used normative orthography is in use. An additional advantage of our approach is that when orthographic systems are replaced, as is currently occurring in Kazakhstan for Kazakh, there is no need to completely rewrite an existing mature transducer; instead, a supplemental transliteration transducer can be constructed.

7 Concluding remarks

The primary contributions of this paper are a wide-coverage morphological description of Crimean Tatar able to analyse and generate both Cyrillic and Latin orthographies, and a general approach to building biscriptual transducers.

Acknowledgements

This work was made possible in part by Google's Summer of Code program, the Swarthmore College Research Fund, and the involvement of the Apertium open source community. We also very much appreciate the thoughtful, constructive comments from several anonymous reviewers.

References

Ekaterina Ageeva and Francis M. Tyers. 2016. Combined morphological and syntactic disambiguation for cross-lingual dependency parsing. In *Proceedings of TurkLang 2016*, Bishkek, Kyrgyzstan.

Kemal Altıntaş and İlyas Çiçekli. 2001. A morphological analyser for Crimean Tatar. In *Turkish Symposium on Artificial Intelligence and Neural Networks (TAINN2001)*.

Lars Johanson and Éva Á. Csató. 1998. *The Turkic Languages*. Routledge.

Lauri Karttunen. 1993. *The Last Phonological Rule: Reflections on constraints and derivations*, chapter Finite-state constraints. University of Chicago Press.

Darya Kavitskaya. 2010. *Crimean Tatar*. LINCOM Europa.

Krister Lindén, Miikka Silfverberg, Erik Axelson, Sam Hardwick, and Tommi Pirinen. 2011. *HFST—Framework for Compiling and Applying Morphologies*, volume 100 of *Communications in Computer and Information Science*. Springer.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Chris Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Dan Zeman. 2016. Universal Dependencies v1: A Multilingual Treebank Collection. In *Proceedings of Language Resources and Evaluation Conference (LREC'16)*.

Gary F. Simons and Charles D. Fennig, editors. 2018. Twenty-first edition edition. SIL International, Dallas, Texas. Crimean Tatar: https://www.ethnologue.com/language/crh.

Jonathan North Washington, Aziyana Bayyr-ool, Aelita Salchak, and Francis M. Tyers. 2016. Development of a finite-state model for morphological processing of Tuvan. *Родной Язык*, 1(4):156–187.

Jonathan North Washington, Ilnar Salimzyanov, and Francis M. Tyers. 2014. Finite-state morphological transducers for three Kypchak languages. In *Proceedings of the 9th Conference on Language Resources and Evaluation, LREC2014*.