Finite-state Relations Between Two Historically Closely Related Languages

Kimmo Koskenniemi
University of Helsinki, Finland
kimmo.koskenniemi@helsinki.fi,

ABSTRACT
Regular correspondences between historically related languages can be modelled using finite-state transducers (FST). A new method is presented by demonstrating it with a bidirectional experiment between Finnish and Estonian. An artificial representation (resembling a proto-language) is established between two related languages. This representation, AFE (Aligned Finnish-Estonian) is based on the letter by letter alignment of the two languages and uses mechanically constructed morphophonemes which represent the corresponding characters. By describing the constraints of this AFE using two-level rules, one may construct useful mappings between the languages. In this way, the badly ambiguous FSTs from Finnish and Estonian to AFE can be composed into a practically unambiguous transducer from Finnish to Estonian. The inverse mapping from Estonian to Finnish is mildly ambiguous. Steps according to the proposed method could be repeated as such with dialectal or older written texts. Choosing a set of model words, aligning them, recording the mechanical correspondences and designing rules for the constraints could be done with a limited effort. For the purposes of indexing and searching, the mild ambiguity may be tolerable as such. The ambiguity can be further reduced by composing the resulting FST with a speller or morphological analyser of the standard language.

KEYWORDS: Finite-State Transducers, Historical Linguistics, HFST, Two-Level Morphology, Foma.

Kimmo Koskenniemi 2013. Finite-state relations between two historically closely related languages. *Proceedings of the workshop on computational historical linguistics at NODALIDA 2013.* NEALT Proceedings Series 18 / Linköping Electronic Conference Proceedings 87: 43–53.

1 Introduction

In historical linguistics one studies two or more languages which are assumed to be related. Among other things, the scholar collects cognate words, postulates the sounds in the proto-language and establishes regular correspondences or so called sound laws according to which the known languages can be derived from the proto-language. For a more detailed description of the comparative method, see (Campbell, 2004, pp. 127–147). When establishing the proto-language and the correspondences, the scholar both compares the known languages, i.e. uses the external evidence and studies the sound alternations within single languages, i.e. uses the internal evidence. Different scholars may disagree upon the details of proto-languages and often there is even a dispute whether a common ancestor can be established at all, cf. (Campbell, 2004, pp. 345–346).

Finite-state rewriting systems like the XFST (Beesley and Karttunen, 2003) and its open source counterpart FOMA (Hulden, 2009) provide basic tools for expressing and implementing the regular sound relations. In addition, there is an open source toolkit HFST (Lindén et al., 2011) based on FOMA, SFST (Schmid, 2005) and OpenFST (Allauzen et al., 2007). HFST provides a programming interface and command line tools including HFST-TWOLC, a compiler for two-level rules. Regular relations are implemented as finite-state transducers (FST) which transform strings to other strings.

In particular, by using FSTs, one may explicitly check whether proposed sound laws produce the correct results or not. Rewrite rules implement more or less directly the phonological rules and a sequence or cascade of such rules can alter the word in the proto-language into a word in the known language.

FSTs have the advantage that the mappings can be readily inverted or combined with other mappings. Thus, once you have designed rules for deriving language A from a proto-language C, you may invert the rules, and see what possible proto-forms any word of A could have. Such an inverted mapping can be composed with the mapping from C to language B to see which words in B could be of the same origin as a word in A. The inverted rules, e.g. the mapping from C to A, tend to be severely ambiguous. Some description of the morphophonological constraints, e.g. of the proto-language C are necessary, in order to reach reasonably bidirectional mappings.

The source of ambiguity is about the same in the traditional historical sound relations and in the approach presented here. This paper carries out a small scale experiment using Estonian and Finnish cognate words. Instead of using the proto-language, an artificial language through alignment is used. The expressing of the redundancy of the proto or artificial language is the way to produce usable reverse relations.

The present work is not statistically oriented. The reader is advised to consult e.g. (Bouchard-Côté et al., 2009; Bouchard-Côté et al., 2013) for approaches which reconstruct proto-languages in an unsupervised manner and construct phylogenetic trees. This work also differs from the data driven approached based on Minimum Description Length (MDL). For more information on such, the reader should consult e.g. the work of Roman Yangarber and his colleagues (Wettig et al., 2012).

The present work relies on the general linguistic knowledge and linguist's intuition. The conceptual framework in this approach differs in certain respects from the widely accepted comparative method. The most notable difference is the use of direct relations instead of the stepwise application of rewrite rules. The use of two-level parallel rules in controlling the ambiguity makes the individual decisions of the linguist less dependent of each other. Testing of two-level rules is easy because the tools pinpoint the contradicting rule and the point where something in the example does not match the rule.

Even if a linguist does not agree with the framework presented below, she or he may use the method for building practical and useful linguistic modules. In particular, the method may turn out to be useful in describing the relation between dialects or closely related languages. One could also think of extending the method to cope with more realistic proto-language deduction and the study of historical linguistics.

2 Toy Balto-Finnic

This paper reports an experiment with two related languages, Finnish (FI) and Estonian (ET) which are understood to derive from Proto Balto-Finnic (PBF). The paper tries to show how the present tools can be used constructively when one needs to relate two languages or language forms which are close to each other.

Exercise data from (Campbell, 2004, pp.59–61) is used for this experiment. The book gives some 84 carefully selected cognate words in Finnish and Estonian. Some examples of the list which are genetically related words are given in Table 1.

PBF	*FI*	*ET*	*Gloss*
mees	mies	mees	'man'
ikä	ikä	iga	'age'
lehti	lehti	leht	'leaf, sheet'
verkko	verkko	võrk	'net'

Table 1: Sample words: Proto Balto-Finnic, Finnish and Estonian

The written forms of Finnish and Estonian words are used here. Thus some identical sounds are represented using different letters, e.g. **y** and **ü** as well as the Finnish **k** and the Estonian **g**. If one would do serious historical linguistics, one would use e.g. symbols from the International Phonetic Alphabet (IPA)[1] for a phonologically motivated representation of sounds. In the present case, the use of the written forms causes little harm as both languages are written almost phonemically. The method is insensitive to the slight differences in the orthographic conventions.

3 Aligning the words

The Proto Balto-Finnic form is also given in the book, but it is not used in the experiment. Instead, the *Finnish and Estonian forms are aligned letter by letter with each other* using general linguistic knowledge. Vowels may match vowels and consonants consonants. Semivowels match semivowels but may even match vowels or consonants. Estonian words tend to be shorter than the Finnish ones, so some letters in one language may correspond to zero in the other. The above four words would give the following alignment:

```
m ie e s          (*mees)
i kg äa           (*ikä)
l e h t iØ        (*lehti)
v eõ r k kØ oØ    (*verkko)
```

[1] See e.g. http://en.wikipedia.org/wiki/International_Phonetic_Alphabet

Single letters in this alignment indicate that identical letters correspond to each other in that position. A pair, e.g. **äa** indicates that **ä** in Finnish corresponds to **a** in Estonian. This alignment can be done in a fairly objective manner so that almost anybody (or at least any linguist) would arrive at the same result. Matching vowels with vowels and consonants with consonants is nearly sufficient as a criterion. In addition, semivowels e.g. **j** may match either consonants or vowels. Deletion and epenthesis must, of course be allowed, if needed, e.g. **iØ** when there is an **i** in Finnish but nothing in the Estonian form.[2]

Next, we treat this representation of the aligned words as a substitute of the proto-language and call it Aligned Finnish-Estonian (AFE). The AFE has clearly many more (morpho)phonemes than the conventional PBF. In our case, the AFE contains all information that the PBF does and some additional information. We may map all our example words of the AFE into Finnish, Estonian or even to PBF e.g. by using a parallel replace rule of XFST or FOMA. The mappings are many-to-one because there are more symbols in the AFE than in any of the real languages or in PBF. For example, the mapping from the AFE into Finnish using XFST or FOMA would be:

```
regex [a∅->a, eõ->e, ie->i, iõ->i, i∅->i, ji->j, kg->k, k∅->k,
 ou->o, oõ->o, o∅->o, pb->p, p∅->p, td->t, uo->u, u∅->u,
 yü->y, yö->y, y∅->y, äa->ä, ä∅->ä, n∅->n, ∅a->0, ∅ü->0, ∅õ->0] ;
```

Here all single letters are unchanged and the second letter of two-letter symbols is dropped. The transformation from the AFE into Estonian drops, respectively, the first letter.

The mapping from the AFE to PBF takes mostly the first of the two letters with some exceptions:[3]

```
regex [a∅->a, eõ->e, ie->e, iõ->e, i∅->i, ji->j, kg->k, k∅->k,
 ou->o, oõ->o, o∅->o, pb->p, p∅->p, td->t, uo->o, u∅->u,
 yü->y, yö->ö, y∅->y, äa->ä, ä∅->ä, n∅->n, ∅a->0, ∅ü->0, ∅õ->0] ;
```

This mapping appears to be surprisingly simple considering the fact that we did not use our knowledge about PBF when constructing the AFE representation. The mapping is simple perhaps because the two languages are so close to each other and because the examples we included did not cover the more complex cases. If we can do the same with a comprehensive set of examples, we might conclude that such a *proto-language only has features which are present* in the known languages. Nothing has then been entered into the proto-language that was not present (in some form) in either of the languages.

4 Internal regularities of the AFE

Writing the rules which transform this kind of a pseudo proto-language AFE into Finnish, Estonian and PBF is trivial, but not very useful as such. The inverted transformations of these,

[2]At this stage, the alignment was made manually. The author also tried a C program made by Måns Huldén which made the alignment automatically using the method of Gibbs sampling. The automatic alignment was almost identical to the manual on, differing only in three words such as Finnish **kansi** vs. Estonian **kaas** where algorithm aligned **k a n:a s i:Ø** whereas my own alignment was **k a Ø:a n:Ø s i:Ø**. The PhD dissertation of Kondrak gives a good survey of the various methods which have been used in aligning and identifying cognate words in related languages, see (Kondrak, 2002).

[3]The exceptions deal with diphthongs in Finnish which correspond to long vowels in PBF. These include the AFE **iõ ie** (these occur only where there is a diphthong **i e** in Finnish and **e e** in PBF), the AFE **uo** (corresponding to a part of **u o** in Finnish and **o o** in PBF) and the AFE **yö** (a part of **y ö** in Finnish and **ö ö** in PBF).

e.g. from Finnish to AFE, are heavily or infinitely ambiguous unless some constraints are applied. Many letters in Finnish or Estonian have multiple possible counterparts in the AFE. Even worse, a deletion (e.g. in AFE-TO-FI) causes an infinite cycle in the inverted mapping (FI-TO-AFE). Thus the raw composition of FI-TO-AFE and AFE-TO-ET would map the Finnish word to its Estonian cognate and a host of other, unwanted forms.

4.1 Filtering excessive possibilities

In order to make the inverse mappings useful, we describe the *regularities which constrain the AFE*, i.e. in what contexts its (artificial) morphophonemes may occur and where they may not. Combining such a filter with the trivial mappings helps in removing many or most of the unwanted strings. Finnish words could be transformed into Estonian words (assuming that they are related) by a composition of three transducers (where **.o.** stands for composition):

```
FI-TO-AFE .o. AFE-FILTER .o. AFE-TO-ET
```

The first is the inverse transducer of the above FOMA replace rule which takes a Finnish word as input and produces a (possibly infinite) set of AFE strings which AFE-TO-FI would map to the input word. AFE-TO-ET is FST for the parallel FOMA replace rule for Estonian (which maps one AFE string into one Estonian word).

The middle transducer, AFE-FILTER is the key component of the present solution. It uses the alphabet of AFE for its input and output. AFE-FILTER never alters anything. Instead, it removes or forbids sequences which do not conform with the regularities. AFE-FILTER is here implemented using two-level rules see (Koskenniemi, 1983; Karttunen, 1993). To compile the rules into FSTs, HFST-TWOLC, the open source two-level rule compiler was used (Silfverberg and Lindén, 2009). The filter uses two kinds of rules, (1) right-arrow rules (=>) which list the contexts a morphophoneme may only occur, and (2) exclusion rules (/<=) which list contexts where a morphophoneme may not occur.

The rules were written by according to normal linguistic intuition using the AFE form of the example words and studying the distribution of each two-letter morphophoneme in turn. Linguistic judgements were based on a text file containing all example words in their AFE representation. This file was searched using simple regular expressions. On an ordinary Unix or Linux machine, one would use the GREP or EGREP program by searching for interesting morphophonemes and selecting only those lines which contain them. The GNU Emacs has an equivalent command (M-x occur) which was actually used for this purpose.

For determining the constraints concerning Finnish **u**, one looks at the distributions of AFE morphophonemes **u**, **uo** and **uØ**. It is easy to see that **uo** only occurs in the first syllable as a part of a diphthong in Finnish (or a long vowel in Estonian). The **uØ** only occurs at the end of the word where it must be preceded by a double consonant or a long vowel or a diphthong. In this way, constraints for each AFE morphophoneme can be formulated. The constraints are quite independent of each other and there is no ordering among them. All constraints must be respected separately and applying one does not make any of the other constraints less or more applicable. (As opposed to rewrite grammars, two-level grammars have neither bleeding nor feeding.)

No claims are made for the completeness or generality of the rule set produced in this experiment, but it works quite nicely for the examples in Campbell's book. Almost all Finnish example words map into just one (correct) Estonian word. In the inverse direction, most Estonian example words produce a set of Finnish words so that one member of the output set is the correct one, e.g. Estonian **haav** is mapped into Finnish **haava**, **haavi**, **haavo** and **haavu** where the first in this list happens to be the correct one.

4.2 Rules for the constraints

The experiment implemented the filter using the HFST-TWOLC two-level compiler. Normally, two-level grammars relate the lexical and surface representations in order to describe the morphophonological alternations in the inflection. Here, we used the rule compiler in an unusual way where only one relevant representation, the AFE, was involved. The alphabet consisted of all letters and so called morphophonemes in the AFE.

```
Alphabet
  a b d e f h i j k l m n o p r s t u v y ü ä ö aØ eõ ie iõ iØ ji kg kØ
  ou oõ oØ pb pØ td uo uØ yü yö yØ äa äØ nØ Øa Øü Øõ ;
```

With this alphabet, definitions for vowels **V** and consonants **C** are needed, and a particular phonological environment **Dbl** which appears to control many features in Estonian. If a double consonant precedes immediately or a double vowel precedes the single consonant, then certain stem final consonants are deleted.

```
Sets
  V = a e i o u y ä ö aØ eõ ie iõ iØ ou oõ oØ uo uØ yö yü yØ äa äØ ;
  C = b d f g h j k l m n p r s t v ji kg kØ nØ pb pØ td ;
Definitions
  Dbl = [V V | C | V Z nØ] C (V C) ;
```

Using these sets and definitions, certain sequences are excluded from the AFE representations. E.g. for **u** we have two constraints each of which is expressed with two rules. The first rule restricts the possible contexts where an AFE morphophoneme may occur and the second excludes other alternatives in such a location. The first constraint is for long vowels in Estonian (and in PBF) which in Finnish are diphthongs. The second constraint controls the deletion of the stem final vowel in Estonian.

```
Rules
...
"uo"  uo    =>  .#. C* _ o ;
"~u"  u    /<=  .#. C* _ o ;
"uØ"  uØ    =>  Dbl _ .#. ;
"~u"  u    /<=  Dbl _ .#. ;
...
```

The first rule simply says that the AFE morphophoneme **uo** can occur only if there are only consonants to the left of it and an **o** immediately to the right. The second rule forbids the occurrence of **u** in this context. The third rule tells that the AFE morphophoneme **uØ** can only occur at the end of the word and if it is immediately preceded by something that matches the predetermined expression **Dbl** given above. **Dbl** essentially represents either a double consonant or a double vowel as expressed in the regular expression formalism used in the Xerox XFST.

The authoring of these filtering rules proved to be easier than expected because good tools were available. The HFST-PAIR-TEST program tests the compiled two-level rules. It uses a test file which was derived from the AFE representation of the set of examples using, among other things, the AFE-TO-FI transducer. The pair test program immediately points out discrepancies between the examples and the newly written rules. A few hours of intensive writing, testing and tuning was sufficient.

Then, FI-TO-AFE, AFE-FILTER and AFE-TO-ET were composed into a single FST. The testing of this with the input of Finnish test words gave almost clean results. Some tuning was necessary, especially because there were some deletions in the mapping from AFE to Finnish. Soon the filter enabled almost clean results in the mapping from Finnish to Estonian (88 results out of 84 input words).

The inverse mapping from Estonian to Finnish was ambiguous and produced some 4.5 results per input word. This ambiguity was almost exclusively due to the deletion of the stem final vowels in Estonian.

The experiment presented here is a toy. It could easily be expanded to cover a larger part of the regular sound relations between Finnish and Estonian. The result would be an interesting tool for the linguist or a linguistically oriented language learner rather than a practical translator between these languages. The non-deterministic mapping from Estonian to Finnish could still be made more unambiguous by using a finite-state speller for Finnish, e.g. OMORFI (Pirinen, 2011). Composing the Estonian to Finnish mapping with such a speller FST would exclude most of the wrong results because the random addition of the stem final vowel mostly produces non-words in Finnish.

4.3 Alternative implementation

It is clear that one could have expressed the whole mapping from Finnish to Estonian using just the surface representations of these languages and one single two-level grammar. In this grammar, the alphabet consists of true pairs instead of the atomic two-letter morphophonemes. In this implementation, the declaration of the alphabet lists units like **ä ö a:0 i:õ**. The rules combine the pair of two-level rules needed for one constraint in the previous approach, each into a single rule e.g.:

```
"u:o"  u:o  <=>  .#. C* _ o ;
"u:Ø"  u:0  <=>  Dbl _ .#. ;
```

Authoring such rules was not essentially more difficult or time consuming than writing the grammar which checked just the AFE consistency. The present experience suggests that the first approach is somewhat easier to manage. The first approach is also more flexible in cases where one would like to permit certain overlapping contexts for distinct pairs (or morphophonemes). The AFE based filter appears easier to extend or modify allows the linguist to choose different levels of strictness. A particular asset of the first approach is the modularity and symmetry of it. It can be used in additional combinations, e.g. as a component for building Estonian to PBF.

5 Potential applications

The result of the experiment is probably not very useful in other ways than that it extends our understanding and that similar tools may be constructed for more concrete and practical applications. Some applications have been considered, including one related with the Corpus of

Old Finnish texts.[4] The frequency list of the corpus is available on line.[5] Below is a sample of the word frequency list of Old Finnish (OF):

```
75      4116    ioca     0,1201 %
76      4074    händä    0,1189 %
77      3952    hänelle  0,1153 %
...
81      3797    ombi     0,1108 %
82      3787    canssa   0,1105 %
83      3775    sijnä    0,1102 %
```

One could process selected samples of OF and align them with the corresponding words of Modern Standard Finnish (MSF). Proceeding according to the principles and steps above, one would get aligned pairs, corresponding rules, and a transducer which converts MSF to OF and vice versa. This OF-TO-MSF could be combined with a list of possible word forms of MSF, which we can readily get from a Finnish morphological analyser in FST form (Pirinen, 2011). In addition to these components, one would probably want to include a list of exceptions where the OF word is not in a regular relation with the MSF word.

6 Deducing the proto-form

Let us return to the example rules which were discussed in 4.2. The following rules correspond to Finnish **ä**. Let us look, how we can reduce the set of corresponding AFE morphophonemes **äa**, **ä** and **äØ** into a smaller and more realistic set.

```
"äa"  äa   => V C+ _ ;
"~ä"  ä   /<= V C+ _ ;
"äØ"  äØ   => Dbl _ .#. ;
"~äa" äa  /<= Dbl _ .#. ;
```

We can see that the two correspondences (**äa** and **äØ**) allowed by these rules are restricted to clearly defined contexts and that the third one **ä** is allowed elsewhere, i.e. it has no context condition which could be expressed in clear linguistic terms. Therefore, **ä** is a good candidate to represent these three AFE morphophonemes. Suppose that we would change the AFE by reducing **äa** and **äØ** into **ä**. Then we have to replace the simple rewrite rule `äa -> a` (from AFE to Estonian) by rules with a context:

```
ä:a <=> V C+ _ ;
```

Similarly, instead of `äØ -> 0` we would have:

```
ä:Ø <=> Dbl _ .#.
```

After these modifications, the mappings still produces the correct results and we may proceed to make further reductions by joining other AFE morphophonemes. As we join the AFE morphophonemes, we build the mappings by replacing one by one the trivial translation rules of AFE-TO-ET and AFE-TO-FI with rules that correspond to the two-level rules of AFE-FILTER. This process stops when no more simplifications of the AFE morphophonemes can be made. The resulting rule sets correspond to the sound laws according to which the present Finnish and Estonian can be derived out of this newly created candidate for the proto-language.

[4] http://kaino.kotus.fi/korpus/vks/meta/vks_coll_rdf.xml
[5] http://kaino.kotus.fi/sanat/taajuuslista/vks_5000_frek.html

7 Discussion

What does a creation like AFE represent and why does it appear to be useful? We pointed out that AFE is linguistically close to the proto-language PBF. Whereas the PBF consists of a normal set of phonemes, AFE has more symbols in it, to the extent that from AFE, one can directly and unambiguously produce Finnish, Estonian and PBF. This extra information proved to be useful when used as a basis for the filtering. The good thing about AFE is that it is simple to produce, and you need not backtrack with your decisions. You simply produce the representation first, and you need not change it. One can automatically produce various versions of the test data: monolingual (FI, ET, AFE), bilingual (FI:ET, FI:AFE, ...). Text versions can be used by the humans and and the FSTs used by the programs.

In spite of the apparent rawness of the AFE representation, it is relatively easy for the linguist to make generalisations using it. A reasonably small set of examples appears to enable the writing quite general constraints. The AFE representation seems to contain most or all the information in the proto-language that is needed for constraining the mappings. The two-level compiler as well as the HFST toolbox and FOMA appear to be extremely useful tools when processing these kinds of language data and relations.

8 Internal reconstruction

A concluding remark is made concerning the two-level framework when representing relations between closely related languages. Suppose that we would have made a morphophonological two-level analyser for Estonian using a similar style as we used in constructing the AFE in this paper. Many Estonian word stems would then have a morphophoneme instead of a vowel at their end. The Estonian word **h a a v** would be represented by its lexical representation such as **h a a v Øa** (because the missing stem final **a** is present in inflected forms) and the Finnish word would be represented as **h a a v ao**. The new kind of AFE would then have a representation like **h a a v ao-aØ**. One can, then, map the Estonian lexical representations more accurately to the AFE representations and therefore, one can map the Estonian words more uniquely into Finnish words.

When one uses morphophonemes in the real languages, one gets even more morphophonemes in this kind of AFE than in the method presented earlier in this paper. Note that using the morphophonemes of individual languages corresponds to the internal reconstruction of the normal historical linguistics. In order to proceed to the phonemes of a realistic proto-language, one establishes the default members of such complex morphophonemes. Thereafter, one can proceed to stepwise establish a proto-language. Filtering the new kind of AFE would be done according to similar methods as above. Similarly the relations between present languages and the proto-language would be constructed according to the principles presented here.

References

Allauzen, C., Riley, M., Schalkwyk, J., Skut, W., and Mohri, M. (2007). Openfst: A general and efficient weighted finite-state transducer library. In *Proceedings of the Twelfth International Conference on Implementation and Application of Automata, (CIAA 2007)*, volume 4783 of *Lecture Notes in Computer Science*, pages 11–23, Prague, Czech Republic. Springer.

Beesley, K. R. and Karttunen, L. (2003). *Finite State Morphology*. Studies in Computational Linguistics, 3. University of Chicago Press. Additional info, see: `www.stanford.edu/~laurik/fsmbook/home.html`.

Bouchard-Côté, A., Griffiths, T. L., and Klein, D. (2009). Improved reconstruction of protolanguage word forms. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 65–73, Boulder, Colorado. Association for Computational Linguistics.

Bouchard-Côté, A., Hall, D., Griffiths, T. L., and Klein, D. (2013). Automated reconstruction of ancient languages using probabilistic models of sound change. *Proceedings of the National Academy of Sciences*, 10.1073/pnas.1204678110.

Campbell, L. (2004). *Historical Linguistics: An Introduction*. Edinburgh University Press, second edition.

Hulden, M. (2009). Foma: a finite-state compiler and library. In *Proceedings of the Demonstrations Session at EACL 2009*, pages 29–32, Athens, Greece. Association for Computational Linguistics.

Karttunen, L. (1993). Finite-state constraints. In *Proceedings of the International Conference on Current Issues in Computational Linguistics, June 10–14, 1991. Universiti Sains Malaysia, Penang, Malaysia*, pages 173–194.

Kondrak, G. (2002). *Algorithms for Language Reconstruction*. PhD thesis, University of Toronto.

Koskenniemi, K. (1983). *Two-level Morphology: A General Computational Model for Word-Form Recognition and Production*. Number 11 in Publications. University of Helsinki, Department of General Linguistics.

Lindén, K., Axelson, E., Hardwick, S., Pirinen, T. A., and Silfverberg, M. (2011). Hfst – framework for compiling and applying morphologies. In Mahlow, C. and Piotrowski, M., editors, *Systems and Frameworks for Computational Morphology 2011 (SFCM-2011)*, volume 100 of *Communications in Computer and Information Science*, pages 67–85.

Pirinen, T. (2011). Modularisation of finnish finite-state language description — towards wide collaboration in open source development of a morphological analyser. In Pedersen, B. S., Nešpore, G., and Skadiņa, I., editors, *Proceedings of the 18th Nordic Conference of Computational Linguistics NODALIDA 2011*, NEALT Proceedings Series, Vol. 11 (2011), pages 299–302. Northern European Association for Language Technology (NEALT).

Schmid, H. (2005). A programming language for finite state transducers.

Silfverberg, M. and Lindén, K. (2009). Conflict resolution using weighted rules in hfst-twolc. In *Proceedings of the 17th Nordic Conference of Computational Linguistics, NODALIDA 2009*, pages 174–181. Northern European Association for Language Technology (NEALT).

Wettig, H., Reshetnikov, K., and Yangarber, R. (2012). Using context and phonetic features in models of etymological sound change. In *Proceedings of the EACL 2012 Joint Workshop of LINGVIS & UNCLH*, pages 108–116, Avignon, France. Association for Computational Linguistics.