# A morphological analyser for Kven

Sindre Trosterud
Kainun institutti

Trond Trosterud
Giellatekno, UiT

Anna-Kaisa Räisänen
Kainun institutti

Leena Niiranen
UiT

Mervi Haavisto
Kainun institutti

Kaisa Maliniemi
Ruija kvenmuseum

**Abstract**

We present a morphological analyser for Kven, a Finnic language spoken in Northern Norway. Apart from the overall view of the analyser, we discuss principles for whether to treat morphological processes like gemination and stem-conditioned suffix variation as lexically, morphologically or phonologically conditioned alternations. The choice is often governed by the grammatical mechanism itself, but in many instances the analysis may be carried out in more than one way and it becomes a choice of where to place the complexity. The article discusses some of the choices we made in this particular case.

## 1 Introduction

The article presents a morphological analyser for the Kven language, i. e. a program that is an explicit model of the Kven grammar, by which the user may either analyse or generate any Kven wordform, and presents both program design choices and some of the usage possibilities.

The article is structured as follows: Section 2 first gives the technical background for the technologies used to create this analyser, and then an overview of the Kven language itself. Section 3 presents how we have implemented this FST, discusses the rationale behind some of our choices, describes our test setup and mentions the applications in which the analyser is used. In section 4 we evaluate the analyser, and section 5 contains a conclusion.





*The 3rd International Workshop for Computational Linguistics of Uralic Languages*, pages 76–88,
St. Petersburg, Russia, 23–24 January 2017. 

# 2 Background

## 2.1 Theoretical background

This analyser is modeled like a finite-state transducer. A finite-state transducer (FST) is an automaton with both an input tape and an output tape, that defines relations between pairs of strings. When made as a specialized finite-state automation (a lexical transducer) it maps lexemes and morphological specifications (in the form of tags) to corresponding inflected forms, and vice versa.

The transducer modeling morphological processes for Kven actually consists of two separate transducers. The first one is a Lexicon Transducer (Morphological) that takes input A = {the lexeme and a string of tags denoting the grammatical analysis of the word form in question}, then the second one is a morphophonological transducer that uses the first transducer's output B = {the stem + trigger and suffixes} as input, and outputs C = {a string of letters, i.e. the word form}. The transducers are composed, so that $A/B * B/C = A/C$ (or in standard notation: $A : B \; .o. \; B : C = A : C$). The transducer is bidirectional, and gives both analysis and generation.

```
A = t a k k i +N +Sg +Gen
B = t a k k i ^WG >  n

B = t a k k i ^WG > n
C = t a k 0 i 0   0 n
```

In practice (as shown above): the lexicon transducer is for concatenative morphology, pairing lexeme and grammatical tags to stem, suffixes and possible trigger symbols. The morphophonological is for global morphological processes like gemination, consonant gradation and vowel harmony.

A morphological transducer may be described as a set of paths running through the decision tree of all possible stems and affix combinations of the language, and combining lemma + analysis string with its corresponding word form. An ideal transducer for Kven will for a given token return all and only the valid analyses, and for a given analysis generate all and only the valid surface forms.

We built these two components of the Kven transducer using the Finite-State Lexicon Compiler (lexc) as the Lexicon Transducer and the Xerox Two-level Rule Compiler (twolc) as the morphophonological one ([2], [6]). The stems + conjugation are written in `.lexc` files grouped by word type where the stems are connected (pointed) to a specific sublexicon which contains the conjugational abstractions of a subset of a word group consisting of words acting similarly in the conjugations. This process

creates what basically amounts to a decision tree of all possible realisations for all of the words added to the analyser.

The analyser is implemented in the Giellatekno & Divvun infrastructure created and maintained at UiT The Arctic University of Norway, cf. [9](`http://giellatekno.uit.no`).

## 2.2 The Kven language

The Kven language *(kväänin kieli)* is a Finnic language spoken by the Kven people in Northern Norway, primarily in the municipalities shown on the map[1] in Figure 1. The language and the Kvens have their historical roots in the areas that today are part of Northern Sweden and Northern Finland. The wave of Kven emigration towards the Norwegian coast continued for several hundred years beginning from the 16th century.

Figure 1: The main municipalities where Kven is spoken


The Kven language is closely related to Meänkieli in Sweden and the Far Northern dialects of Finnish. However, the language differs from Meänkieli and Finnish dialects because of its separation from these languages e.g. through morphological innovations ([7]) and close contact with Norwegian and North Saami.

As a result of this Kven is a language phonologically very close to northern Finnish dialect, but with several morphological innovation. While sharing its basic vocabulary with Finnish and Meänkieli, Kven has not adapted the vocabulary created for standard Finnish in order to cope with the modern society, instead Kven has borrowed from Norwegian, and to a limited extent also from North Saami. As a result,

[1]The map was made by Alphaios at Wikimedia Commons.

a Finn understanding Finnish dialects and Scandinavian will understand Kven, but a Kven speaker will have greater problems understanding standard Finnish.

Kven is an agglutinative language, which means that to express tense, mood, number, and person, affixes are attached to the word stem. Kven has acquired many loan words not only from Norwegian and Swedish but also from the Saami languages. Many Swedish and Saami loan words already belonged to the language of Kvens when they moved to Norway, because Saami and Swedish loan words can also be found in northern dialects of Finnish and in Meänkieli. Still, the contacts between Kven and Saami speakers were intensified in Norway. The vocabulary of modern life has for the most part been borrowed from Norwegian, cf. [8] for a discussion.

The Kvens have undergone an assimilation process to Norwegian language and culture from late 19th century to late 1940's. The use of the Kven language was forbidden in schools and government offices. After the Norwegianization process the Kven language communities have gradually undergone a language shift, and most of the traditional areas now are monolingual Norwegian. The estimates of the number of speakers of Kven language vary from 2,000 to 8,000, depending on the criteria and methods used.

For political and historical reasons, the Kven language received the status of a minority language in 2005 within the framework of the European Charter for Regional or Minority Languages. Kven was recognized as a language, and not only as a dialect of Finnish. The national minority languages in Norway are protected by ECRML, which opens for protection on two levels and Kven language has been granted the lowest protection level, level II. The lowest level of protection obliges Norway to recognize the minority languages as an expression of cultural wealth, to promote and to protect them, to obtain forms and means for teaching and studies of minority languages, and to promote research of minority languages. Despite of these actions the language is still critically endangered.

The government's current heritage language policy is aimed at strengthening the Kven language in society. These actions include establishing the basic linguistic infrastructure for the language with the descriptive grammar, written standard and dictionaries. The work has been managed mainly by the Kven Institute, the national institution for the Kven language in Norway since 2007. Although the financial resources have been limited, the grammar of the Kven language was published in 2015.

# 3 Modeling phonology or morphophonology

## 3.1 The analyser

The Kven grammatical analyser is based on Eira Söderholm's descriptive grammar of Kven, *Kainun kielen grammatikki* [11], more specifically, the Porsanger variety of Kven (at least for the moment). The lexicon was originally taken from the dictionaries of Terje Aronsen [1] and Eira Söderholm [10], but it has later been completed by words taken partly from Kven informants and partly from corpus texts (`http://gtweb.uit.no/f_korp/`). An overview of the vocabulary is given in table 1.

Table 1: The vocabulary distribution across parts of speech

| Nouns | Verbs | Adj | Closed | Names | Total |
|---|---|---|---|---|---|
| 5,300 | 2,500 | 850 | 1,200 | 30,0000 | ≈ 40,000 |

The vocabulary is divided into 12 lexc files each consisting of all the lemmas, with stems and sublexica classification for each specific word type, as e.g. the entry for *amerikkalainen* 'American':

```
amerikkalainen:amerikkalai n_42 ;
```

The lexicon files contains 93 sublexica for categorizing the different stem types of the open parts of speech (34 for adjectives, 39 for nouns, and 23 for verbs). The parts of speech partly share the sublexica for representing derivational and inflectional suffixes, there are 210 such lexica.

In addition to the lexicon files there is a separate twolc file containing all the morphophonological changes possible within the stems (gemination, consonant gradation, and vowel shortening). This file includes the code for stem-final vowel or consonant changes, and for suffix changes like vowel harmony. The code compromises of 62 rules, with 128 context definitions for triggering.

The tagset is deliberately kept within mainstream Uralic descriptive linguistics, which means that it quite close to the bulk of the *Giellatekno* analysers. It deviates to a certain extent from the Omorfi analyser for Finnish, which is closer to the newer *ISO suomen kielioppi* (Finnish has e.g. `+InfA` and `+InfMa` where Kven has `+Inf` and `+Inf3`), but overall, the tagsets may be converted to each other.

## 3.2 Consonant gradation and vowel harmony

In the Finnic languages, consonant gradation (= cg) is seen as a process concerning the plosives *k, p, t* The most concise presentation of the Finnish system to date is [5]). The same view has been held on other Finnic languages. A notable exception has been Söderholm´s standard grammar of Kven. Here consonant gradation (*graadivaihettelu*, op. cit. p.64ff) covers all the consonants *đ, h, j, k, l, m, n, p, r, s, t* and $v$. [11]. Söderholm unifies consonant gradation and what in Finnish dialectology has been known as gemination, and thus gets a much broader gradation pattern, for more consonants, more triggering contexts , and even an additional grade (e.g. *kk:k:j* for *lukkeet : lukenu : lujen* 'to read, (has) read, I read´) for alternations that in Finnish dialectology has been seen as gradation ($k : j$) and gemination ($k : kk$). The stem for $lukkeet$ in the system is thus $luke$. Söderholm knowingly treats gemination and consonant gradation as the same, while we don't. These two processes have different contexts and different realizations making it only logical from a technical point of view to keep them separated.

Gradation is confined to *p, t, k*, whereas all consonants participate in gemination. The process itself is simpler for gemination (a short consonant gets lengthened) than for consonant gradation (which includes both quantitative and qualitative patterns). Historically, consonant gradation was originally a phonological process: Long unvoiced geminates were shortened and short unvoiced ones were voiced in front of closed syllables, *puku : puvun* 'coat'. This does not hold as a generalization in the contemporary language, and consonant gradation is thus in transducers for Finnic and Saami languages encoded as triggered by a special symbol ^WG inserted in the lexc continuation lexicon rather than by referring to the closed syllable, like here for $k : v$ (the left and right vowel contexts are specific in order not to overlap with the $k : j$ and $k : 0$ gradation contexts. Also an example how inclusion is easier than exclusion in twolc rules, as later mentioned in more details.)

```
"Gradation k:v"
k:v <=> [Vow - i] _ [o|ö|u|y] (:i) ^WG:0 ;

!! Test for this example:
!!€ puku^WG>n
!!€ puvu0>n
```

As a contrast, consider vowel harmony, which is treated as a phonological process. Consider the slightly simplified rule "Back harmony" below, where `NonFront` is defined as any segment not in the set e i y ä ö ü æ ø, and `BackVowel` is defined as a member of a o u å.

```
"Back harmony"
 Vx:Vy <=> BackVowel: NonFront:*  _ ;
         where Vx in ( %^A %^O %^U )
               Vy in (   a   o   u )
         matched ;
```

### 3.3 Gemination as insertion of consonant

As seen in the previous section, we treat consonant gradation morphologically (weak grade is triggered by an explicit symbol ^WG rather than by a phonological context), but vowel harmony we treat phonologically. When it comes to consonant gemination, we are facing the same choice.

In Kven, a short consonant following a short stressed vowel becomes lengthened when followed by a long vowel, like the nominative : partitive pair *sana* : *sannaa* 'word'. The lexc representation of the partitive form is `sana>^V` (i.e. the stem *sana*, a suffix boundary and the partitive suffix ^V, or vowel copy. We have chosen to treat this as a phonological process. The consonant gemination is thus treated in twolc as a rule that doubles a given consonant in the context defined by the rule. For example as follows:

```
"Consonant gemination for a"
0:Cx <=> Cx _ :a %>  :a ;
where Cx in  {đ h j k l m n p r s t v} ;
```

An alternative treatment would have been to avoid consonant insertion, by operating with a gemination place-holder in the stem instead, and by operating with a trigger instead of relying upon the second syllable long vowel acting as a trigger. In the alternative rule below, a dummy ^RC (root consonant) is changed to a copy of the preceding consonant whenever followed by a lengthening trigger ^CNSLEN.

```
"Consonant gemination for a (alternative approach)"
%^RC:Cx <=> Cx _ ... %^CNSLEN: ;
```

In the lexicon file, this alternative approach would have been implemented as follows: The gemination place-holder would have been added to all stems containing a single root consonant, and all suffixes introducing a long second syllable would have been enriched with the trigger symbol as well:

```
LEXICON Nouns
sana:san^RCa n_21 ;
```

```
...
LEXICON n_21
+Par:^CNSLEN%>^V K ; !  %^V = vowel copy
```

Our approach gives a more readable lexicon (the stem being *sana* rather than *sanˆRCa*, at the expense of running the risk of introducing gemination in contexts where it should not have been.

One might argue that using explicit triggers from lexc to twolc make the rules easier to define (as the triggers are unique) and more easily maintainable than "implicit triggers" that try to work with the letter itself.

## 3.4 Testing

We built a YAML test suite of hand-made conjugation paradigms for 54 nouns, 17 adjectives, 39 verbs and 2 pronouns, totaling 112 words. They where chosen so that we at least had one representative for each of groupings of words in the reference grammar [11]. These paradigms amount to 8,064 analysis and generation tests.

Since twolc rules are "global" in the sense that they actualize any time circumstances of the given context are fulfilled there is naturally a risk of false positives, where there are other unforeseen circumstances that also match the rules' contexts and the rule actualizes even though it shouldn't. Another challenge is when rules have contexts that don't seem to overlap, until a word form comes up that fits into both. There is a trade-off between having redundant, but clear and explicit lexc code (increased amount of sublexica); and short and concise twolc where the complexity is in narrowing down the contexts for the rules to only actualize in the instances you want.

Creating a rule that matches a certain context is much simpler than making sure there are no other conceivable contexts that also trigger the same rule. In other words, including the context you want is easy, excluding the rest is difficult.

## 3.5 Practical applications

The analyser is already in use in several related projects. It has been used in the creation of a morphologically enhanced e-dictionary (`http://sanat.oahpa.no/fkv/nob/`) and especially it's "point and click" functionality for translating words on any website (see [3] for a presentation of the Kven dictionary, and [4] for a presentation of the dictionary platform), a Kven speller that may be downloaded for use in LibreOffice or MS Word, and the publicly available e-learning tools Oahpa (`http://oahpa.no/kveeni`).

# 4 Evaluation

In the evaluation section, we look at lexical coverage, and discuss problems with text coverage for Kven. We then look at the grammatical coverage, i.e. to what extent it is able to analyze and generate the word forms of a representative set of Kven inflectional paradigms.

## 4.1 Evaluating text analysis

Kven is an *Ausbau* language, which means that from a former stage where Kven was not distinguishable from other Northern Finnish dialects, both Kven and Finnish have developed in different directions. Contemporary Kven writers are influenced by Finnish, and the Kven written norm that forms the foundation of the present analyser has a weak written tradition.

In order to evaluate the analyser, we tested it against the full Kven corpus[2], and against two subparts of it, one strictly adhering to the norm, and one consisting of texts taken from the Kven newspaper *Ruijan Kaiku*. For each of the 3 text collections, we give the coverage of the analyser, and for the words not recognized by the analyser, we indicate how large a part of these missing words are recognized by a Finnish analyser.[3]

Table 2: Coverage on different parts of the available corpus

| Text type | Words | Coverage | Finnish OOV |
|---|---|---|---|
| Corpus adhering to norm | 56,116 | 88.4 % | 37.5 % |
| Contemporary news text | 4,212 | 79.3 % | 44.0 % |
| The full Kven corpus | 260,375 | 84.9 % | 40.1 % |

The lowest coverage we get from the news material, the Kven periodical *Ruijan Kaiku*. Several of the words missing in the analyser are neologisms taken from the news domain, such as *minoritetti* 'minority', *standaarttii* 'standard', *elästytethään* 'revitalizes', while the dictionaries upon which the analyser was built have to a larger degree been geared towards the traditional language, as this is what most of the corpus has included. The newspaper was written in Finnish several years before starting

[2]The corpus is available at `http://gtlab.uit.no/f_korp/`

[3]The analysers are the ones found at http://victorio.uit.no/langtech/trunk/langs, the svn version used is r143378.

to write in Kven, and it is thus also no surprise that it has the highest percentage of Finnish among the words not known to the Kven analyser.

The coverage for the corpus adhering to norm is better, but still below 90 %. The reason for this is twofold. First, the text in question is a grammar ([11]), and as such it contains both scientific terminology, neologisms, affixes and dialect forms outside the standard. Second, the analyser still represents work in progress, and parts of both the morphophonology (especially the interplay between gemination and stem vowel changes linked to the $-i-$ suffix) and derivation are still not accounted for.

The non-recognized part of the corpus was also tested against a Norwegian analyser. We do not include any numbers for this, since the data was contaminated by conversion errors and citation loans, but the share of unassimilated Norwegian loans in the Kven text itself was small.

## 4.2 Evaluating grammatical paradigms

In order to test the morphological performance of the analyzer, we used the 8,064 word forms with their accompanying grammatical analyses, as described in section 3.4 above. The analyzer was tested both against analyzing and generating the set of word forms, and it returned 7,962 passes, or a correct percentage of 98.7 %.

Looking at the grammatical classification of Kven words found in the standard reference grammar [11], there are more tests than there are paradigm types in the grammar. In these tests, having at least one test set per LEXICON makes sense in the software engineering and unit test type of way, and having more than one examples makes it linguistically interesting. Therefore some classes are more heavily represented, the most obvious example of this over-representations are the continuation lexicon `n_21` for which we have 7 yaml files, `n_21ie` for which we have 3, and `n_31si` for which we have 5. All these belong to the same class (nominal stem class 2.1 in the grammar), and (not coincidentally) these are some of the sublexica with the most amounts of words to them.

This over-representation should not have a significant impact on this tests performance, since the 3 lexemes with the most errors have 7 errors out of 159 inflectional forms, 6 out of 160 and 5 out of 159 errors. This implies that the errors are spread out evenly across the paradigms, so even if assuming the duplicated test (multiple tests for one class) where testing those we do better, the skew would be minor.

An example of a test suite is shown below.

```
[ 1/25][PASS] paivukko+N+Sg+Nom => paivukko
[ 2/25][PASS] paivukko+N+Sg+Gen => paivukon
...
```

```
[23/25][PASS] paivukko+N+Pl+All => paivukoile
[24/25][FAIL] paivukko+N+Pl+Ess => Missing results: paivukoina
[24/25][FAIL] paivukko+N+Pl+Ess => Unexpected results: paivukkoina
[25/25][FAIL] paivukko+N+Pl+Com => Missing results: paivukoine
[25/25][FAIL] paivukko+N+Pl+Com => Unexpected results: paivukkoine
----------------------------------------------------------------
[ 1/25][PASS] paivukko => paivukko+N+Sg+Nom
[ 2/25][PASS] paivukon => paivukko+N+Sg+Gen
...
[23/25][PASS] paivukoile => paivukko+N+Pl+All
[24/25][FAIL] paivukoina => Missing results: paivukko+N+Pl+Ess
[25/25][FAIL] paivukoine => Missing results: paivukko+N+Pl+Com

Total passes: 46, Total fails: 4, Total: 50
```

# 5 Conclusion

In this article, we have presented an analyser for Kven. Although still containing flaws in both morphophonology, morphology and the lexicon, the analyser has a coverage of close to 90 % on normative Kven text, and has proved to be good enough to provide value when put to use in practical applications.

Linguistically, the stem list is kept maximally simple. This has made it easier for people without technical familiarity of the project to improve the lemma list. Consonant gradation is marked with a trigger in the morphology. The complicated system of root syllable consonant gemination is treated as a phonological process, so that a copy of the preceding consonant is inserted whenever the following vowel is long.

Kven as a standardised language is just in the making, with very few users adhering to the explicit written standard, a standard which is still not finalized. As a result of this, few text corpora may be seen as representing the norm, and language technology tools must be buildt with explicit normative statements, rather than upon analysing texts. A pleasant side effect of this it that the end result proves useful in a wide range of contexts.

With this set of architectural choices the Kven analyser is somewhat different from FST analysers for other Finnic and Saami languages, thereby offering an alternative approach to handling complex morphological properties of this type.

## Acknowledgements

Thanks to Pirjo Paavalniemi for participating in initial work on the analyser, to Sjur Moshagen for building the infrastructure, to Lene Antonsen for comments to the twolc rules, and Hilde Skanke for providing a home for the project at Kainun Institutti. And last, but not least, we wanto to Eira Söderholm for writing a grammar we can build upon and for being available to discuss tricky nuances with the analysis.

## References

[1] Terje Aronsen. *Kvensk-norsk-kvensk elektronisk ordbok*. Universitetet i Tromsø, 2010.

[2] Kenneth R. Beesley and Lauri Karttunen. *Finite State Morphology*. Studies in Computational Linguistics. CSLI Publications, Stanford, California, 2003.

[3] Mervi Haavisto, Kaisa Maliniemi, Leena Niiranen, Pirjo Paavalniemi, Tove Reibo, and Trond Trosterud. Kvensk ordbok på nett - hvem har nytte av den? *Skrifter / Nordisk forening for leksikografi*, 13:176–192, 2014.

[4] Ryan Johnson, Lene Antonsen, and Trond Trosterud. Using finite state transducers for making efficient reading comprehension dictionaries. In NEALT Proceedings Series, editor, *Proceedings of the 19th Nordic Conference of Computational Linguistics*, volume 16, pages 59—71, 2013.

[5] Fred Karlsson. *Suomen kielen äänne- ja muotorakenne*. WSOY, Juva, 1983.

[6] Kimmo Koskenniemi. *Two-level Morphology: A General Computational Model for Word-form Production and Generation*, volume 11 of *Yleisen kieliteteen laitos*. Helsingin yliopisto, Helsinki, 1983.

[7] Anna-Riitta Lindgren. *Miten muodot muuttuvat. Ruijan murteiden verbitaivutus Raisin, Pyssyjoen ja Annijoen kveeniyhteisössä*. Universitetet i Tromsø, Tromsø, 1993.

[8] Anna-Riitta Lindgren and Leena Niiranen. *The Morphological Integration of Scandinavian and Saami Verbal Borrowings in Kven and Their Impact on Contact-induced Language Change*. SKS, 2016.

[9] Sjur Moshagen, Jack Rueter, Tommi Pirinen, Trond Trosterud, and Francis M. Tyers. Open-source infrastructures for collaborative work on under-resourced

languages. In *Collaboration and Computing for Under-Resourced Languages in the Linked Open Data Era*, LREC, pages 71–77, Collaboration and Computing for Under-Resourced Languages in the Linked Open Data Era.

[10] Eira Söderholm. *Kainun sana- ja sanahaamulista Aikamatkaa varten.* Kvenfolket, http://kvenfolket.origo.no/-/page/show/2937_kvenskordliste, 2009.

[11] Eira Söderholm. *Kainun kielen grammatikki*, volume 1408 of *Suomalaisen kirjallisuuden seuran toimituksia.* SKS, Helsinki, 2014.