Alignment and Two-level relations

Kimmo Koskenniemi

Two-level idea (1981): comparing strings

- Rules compare aligned string pairs directly (eg. for vowel harmony)
- Strings are aligned by adding zero symbols (Ø)
- The same alignment (adding of Ø:s) for all rules
- Individual rules accept or reject aligned string pairs
- Rules are unordered (or parallel)
- Ungrammatical string pairs are rejected by some rule
- Grammatical string pairs are accepted by all rules
- No intermediate representations between the two strings
- Ø is not an empty string (ε)

Applications

- Compare underlying representations of words with their surface forms
  ⇒ Morphological analyzer
- Compare cognate word pairs of the two related languages
  ⇒ Phonological mapping between closely related languages and building proto-languages (i.e. historical linguistics) see: Koskenniemi, 2013, *Finite-state relations between two historically closely related languages*
- Compare forms of present language with their old (or dialectal) forms
  ⇒ Indexing of historical texts (or texts written in dialects)

Comparing cognate words

- L. Campbell. *Historical linguistics*, 2004, pp.59-61. Exercise 2.7, Sound change - Balto-Finnic
- 84 pairs of Finnish–Estonian cognate words, eg.
  kala:kala, **juoni:joon**, **lauta:laud**, **velka:võlg**
- Aligned: **kala:kala**, **juoni:joonØ**, **lauta:laudØ**, **velka:võlgØ**
- Character correspondences: **k:k**, **a:a**, … **t:d**, **e:õ**, **i:Ø**, **a:Ø**
- Test set of these 84 aligned word pairs (as char pairs: **v eõ l kg aØ**)
- A rule for each non-identical pair, e.g. **"aØ" aØ => [V V|C] C _ .#. ;**
- Test each rule immediately against the test set
- Rules worked soon (one intensive afternoon)

Finnish-Estonian cognates as strings of character pairs

pairs	*FI align*	*FI*	*ET*	*ET align*
j uo o n iØ	juoni	juoni	joon	joonØ
k a l a	kala	kala	kala	kala
k a l m aØ	kalma	kalma	kalm	kalmØ
k a r ji aØ	karja	karja	kari	kariØ
k a Øa nØ s iØ	kaØnsi	kansi	kaas	kaaØsØ
k ie e l iØ	kieli	kieli	keel	keelØ
k i r p pØ uØ	kirppu	kirppu	kirp	kirpØØ
k i v i	kivi	kivi	kivi	kivi
k oõ r pb iØ	korpi	korpi	kõrb	kõrbØ
k uo o r iØ	kuori	kuori	koor	koorØ
k u r kg iØ	kurki	kurki	kurg	kurgØ

Strings of pairs contain all information

From:

```
k a r ji aØ
```

one can readily and uniquely deduce both the Finnish form:

```
k a r j a
```

and the Estonian form:

```
k a r i
```

If we know in what contexts each pair may occur, we have found the phonological relation between Finnish and Estonian

Studying each character pair

```
$ egrep 'j' afe-given.text
```

(or Emacs Occur)

```
j a l kg aØ
j oõ kg i
j uo o n iØ
k a r ji aØ
m a r ji aØ
o r ji aØ
p oõ h ji aØ
```

Deducing rules

```
j a l kg aØ
j oõ kg i
j uo o n iØ
```

```
k a  r ji aØ
m a  r ji aØ
  o  r ji aØ
p oõ h ji aØ
```

```
a r _ aØ
o r _ aØ
oõ h _ aØ
```

```
"ji"  ji  => [h|l|r] _ aØ .#. ;

"j"   j  /<= [h|l|r] _ aØ .#. ;
```

Discovering further cognates

- 28 081 nouns from a 15 716 free word list (Nykysuomen sanalista, Kaino, Kotus)
- 15 716 nouns from an Estonian free lexicon (*tyvebaas.txt* from eki.ee)
- Compared these lists using the rules (finite-state composition)
- 1525 matches: **...pahka:pahk, pahna:pahn, paja, pajatso:pajats, paju, ...**
- Evaluated a random sample of 100 pairs out of these 1525
- ¾ of the sample appeared to be similar in meaning (due to common genetic origin or borrowing from a common source)

Processing of Old Finnish: Biblia

Ne caxi padzasta / sen yhden meren / ne caxitoista kymmendä waskihärkä jotca seisoit jalcain alla / jotca Cuningas Salomo oli andanut tehdä HERran huoneseen / näistä astiosta carttui ylönpaltinen waski . [B1-Jer-52:20-412c]

A part of "Biblia", the Finnish translation of the Bible from 1642. A sample of 270 k words from the version available in the Kaino service of Kotus.

Possible ways to normalize

waskihärkä	**jotca**	**seisoit**	**jalcain**	**alla**	*old forms*
vaskihärkää	jotka	seisoivat	jalkojen	alla	*modern forms (translation)*
vaskihärkää	jotka	seisoi**t**	jalka**in**	alla	*modern with old affixes*
vaskihärkä	**joka**	**seisoa**	**jalka**	**alla**	*modern base forms*

Steps in relating Old and Modern Finnish

- Pirkko Kuutti from Kotus collected sample of 159 words which represent the principal differences between the Old and Modern Finnish.
- Each old word was paired with its modern counterpart, e.g. **…, nijsä:niissä, nimes:nimesi, nosnut:noussut, näen:näen, näköns:näkönsä, ...**
- Pairs were aligned (first manually, in later steps automatically):
  n i i:j s:Ø s ä , n i m e s i:Ø, n o u:Ø s s:n u t , n ä e n, n ä k ö n s ä:Ø
- Rules were written and immediately tested for each pair, e.g.
  i:j => i _ ;
- Combined with OMORFI in order to reduce ambiguity

Evaluation of the Biblia processing

- In a sample of 97 old tokens, 89 got a correct modern base form
- On the whole, the success rate was good
- Some misses could not be fixed in any elegant way
- Old Finnish uses some different endings e.g. **seisoit** instead of **seisoivat**
- It is very tedious to change endings in OMORFI (as the endings are repeated in a few hundred places, Antti Kanner has tried)
- Compounding in OMORFI is uncontrolled (and produces noise results)

Design goals of OMORFI

- Combining affixes only according to unmarked modern usage is necessary for translation into Finnish, but restricts processing even texts from 1900s not to mention earlier or dialectal texts.
- Maximum coverage (lots of names and slang words and unrestricted compounding) is good for spelling checking but degraded the filtering of the candidate words that the rule component produced out of the Biblia words.
- Free compounding in OMORFI increases the coverage but allows incorrect matches in the filtering.
- Easy to maintain modern vocabulary in OMORFI but difficult to modify the morphological grammar.

Written word forms were not enough

- **nägyn:näØön** as in "*... andoi monelle sokialle nägyn*"
- Allowing Modern Finnish **Ø** to correspond to **g** in Old Finnish creates lots of ambiguity
- Better: Modern Finnish **k:Ø** alternation corresponds to **g** in Old Finnish

- Estonian-Finnish: **jalgØ:jalka** etc.
- Allowing Estonian word-final **Ø** to correspond to **a** in Finnish also ambiguous
- Better: Estonian **Øau** alternation (jalg-jalga-jala-jalgu-jalu) corresponds to Finnish **a** (or perhaps **ao** alternation)

Alignment

- Weighted finite-state transducers can quantify the discrepancy between sounds (or letters)
- A small Python script can produce the weighted transducer
- A small Python script can perform the alignment
- For the Old Finnish examples, the automatic alignment worked fine when adding further examples
- The positioning of zeroes is vital – poor alignment leads to more complicated rules

Defining distances between sounds with Python: A sample from a small program which computes and writes *chardist.fst*

```
vowels = {
    i:(Close,Front,Unrounded),
    y:(Close,Front,Rounded),
    u:(Close,Back,Rounded),
    e:(Mid,Front,Unrounded),
    ö:(Mid,Front,Rounded),
    o:(Mid,Back,Rounded),
    ä:(Open,Front,Unrounded),
    a:(Open,Back,Unrounded)}
```

```
import sys, io, fileinput, hfst
align = hfst.HfstInputStream("chardist.fst").read()
for line in sys.stdin:
    (f1,f2) = line.strip().split(sep=":")
    w1 = hfst.fst(f1)
    w1.insert_freely(("Ø","Ø"))
    w2 = hfst.fst(f2)
    w2.insert_freely(("Ø","Ø"))
    res = w1.compose(align)
    res.compose(w2)
    res.n_best(1)
    paths = res.extract_paths(output=text)
    print(paths.strip())
```

Writing rules from aligned cognate words

- If the alignment is correct, it is easy to write the rules
- If it is poor, it is more difficult to write the rules and the rules will be uglier (and it is best to revise the alignment or the examples)
- No creative imagination is needed, just some linguistic training and the selection of good examples or cognate words
- The set of rules is fully determined by the alignment
- The examples provide all necessary information for formulating the context part in the rule

Re-implementation of Finnish morphological analyser

Goals:

- Flexibility - ability to use for various purposes
- Morphophonemes according to strict alignment of stems and endings
- Easier to modify endings
- Coverage - all forms accepted, not only the common and unmarked ones
- Initially only the central lexicon

The set of stems defines the underlying representation

Segmented stems of 'tuntea' (the only in its infl. class in KSK 14, NSSL 59):

t u n t e (vat) , t u n n e (n) , t u n s (i) , t u n t (isi)

Aligned stems of 'tuntea':

t u n t e , t u n n e , t u n s Ø , t u n t Ø

Lexical representation:

t u n {tns} {eiØ}

What these morphophonemes, eg. {tns], are?

- Very little opinion or creative intuition involved, therefore they are more like facts than hypotheses (morphophonemes are just a list of the alternatives)
- Not phonemes – there are more morphophonemes than phonemes
- They carry information of inflectional patterns
- They make the writing of the rules easier because morphophonemes are more specific (rules need not deduce which alternation applies, only how this alternation must be realized here)
- More rules are needed but they are simpler
- Sometimes the morphophonemes carry information about the origin of a word, e.g. how old a loan-word is an from which language it comes from.

Only few inflectional classes are needed

saartaa:saar{trs}{aoØe} /v;

laskea:lask{eiØ} /v;

lukea:lu{kØ}{eiØ} /v;

tuntea:tun{tns}{eiØ} /v;

potea:po{tds}{eiØ} /v;

(some 30 inflectional classes are collapsed into a single class)

The rules now produce surface forms from the base forms

lukea+V+POTN+ACT+3SG:lukenee

lukea+V+POTN+PSS+4PE:luettaneen

lukea+V+PRES+ACT+1SG:luen

lukea+V+PRES+ACT+3PL:lukevat

lukea+V+PRES+ACT+3SG:lukee

lukea+V+PRES+PSS+4PE:luetaan

Also from the morphophonemic representation

lu{kØ}{eiØ}{nlrs}e{VØ}:lukenee

lu{kØ}{eiØ}{nlrs}{uy}t:lukenut

lu{kØ}{eiØ}n:luØen

lu{kØ}{eiØ}v{aä}t:lukevat

lu{kØ}{eiØ}{VØ}:lukee

lu{kØ}{eiØ}{Øt}t{aä}ess{aä}:luØettaessa

lu{kØ}{eiØ}{Øt}t{aä}isi{V}n:luØettaisiin

Entering new lexemes into the lexicon

When one has the initial lexicon working, one can prepare an open lexicon where lexemes are replaced by regular expressions which represent roughly the possible stems.

```
<Cons Vow+ [l|r|n] %{eØ%}> /v ; ! V25-27 kuolla,purra,mennä
```

A small Python script helps the user in finding the correct inflectional class

```
Enter forms of the next lemma
puhkaisevat
  remaining possible entries =
    puhkaisev{aoe} /v
    puhkaisev{aØe} /v
    puhkaise{pv}{aoe} /v
    puhkaise{pv}{aØe} /v
    puhkais{eØ} /v
puhkaissut
  remaining possible entries =
    puhkais{eØ} /v
entry now fully determined
```

Harvesting lemmas from corpora

The guesser could be modified so that it processes lists of word forms extracted from a corpus:

- Individual forms typically get several proposed lemmas.
- Some lemmas are proposed from several word forms in the list.
- Lemmas which outperform its competitors are selected, i.e. a lemma wins if it covers all forms its competitor does and in addition some more.

From a big corpus one might get lots of good candidate lexemes for the expansion of the lexicon. (Some tuning of the formulas is needed.)

Conclusions

- A set of well chosen examples is valuable
- Alignment is useful and important
- Alignment implies underlying forms mechanically
- Morphophonemes so produced are not phonemes
- Alignment and the examples determine what rules need to be written
- Rules can be tested against the examples right away
- Authoring the rules is easy (maybe an algorithm could do it as well)
- Morphophonemes defined by alignment carry information about etymology
- Two-level rules can be used in some new ways