A typological view of analogy in morphology: some issues and possible solutions

Matías Guzmán Naranjo

1-4.09.2022

Analogy in inflection: the state of affairs

It is hard to evaluate where we are at as a field, because:

- There are many different definitions of analogy
- There is no unity in our goals
- There is no unity in our core assumptions

Analogy in inflection some issues

Why "we" like analogy-based models:

- Simpler architecture
- Fewer weird assumptions
- Certain inflectional patterns are easier to capture with proportions

Motivation of this talk

Most work on analogy in inflection has heavily focused on affixal patterns.
However, there are other types of inflectional patterns rarely treated explicitly:

- Reduplication (Nahuatl, Latin, Persian, ...)
- Metathesis (Russian, Czech, ...)
- Harmony (Hungarian, Turkish, ...)
- Suprasegmental/tonal/length patterns (Russian, Kasem, Amuzgo)
- Free-morph-order (Chintang)
- Morph-positions (Swahili)
- etc.

These are trickier...

Why formalisms?

Because:

- we need certainty that our models work
  - does the analysis actually capture the facts?
  - does the analysis interact well with other parts of the system?
  - does the analysis make testable predictions about unseen data?
- we need to be able to implement our models computationally
  - linguistic systems are massive, humans cannot evaluate analyses by hand.
  - testing many languages becomes impossible
- we need to be able to induce our models automatically

How formalisms?

Things to consider:

- Minimum complexity
- Generative power
- Implementation
- Automatic induction

Juggling these can be tricky.

Which formalism?

Formalisms in analogy are not new:

- X-notation (?, also some 90s computational linguistics work)
- String unification (Calder)
- X-notation improvement (Beniamine)
  - well implemented
  - can handle more complex patterns
  - fast
  - induction
- HPSG-based, relation append implementation (Guzmán Naranjo)
  - well implemented
  - can handle **any** pattern
  - **very** slow (it's TRALE!)
  - no induction
- ...

Proportional analogies I

There are several proposals for writing proportions:

- cant*o* :: cant*aba*

- Xo $\rightleftharpoons$ Xaba (from the traditional literature)
- o $\rightleftharpoons$ aba / t_ (Bonami and Beniamine 2016)

These scale poorly.

Proportional analogies II

For example:

1. carta :: casta
2. marbarpo :: marbaspo

Based on (1), we could postulate:

- XrY $\rightleftharpoons$ XsY
- r $\rightleftharpoons$ s / _ta

However, neither would work correctly on (2)

Proportional analogies III

Another example:

- carta :: catra

This can't be expressed with either approach:

- XYZW $\leftrightharpoons$ XYZW

Does not even work when reapplied to the same alternation:

- carta $\rightharpoonup$ catra, ctara

Proportional analogies VI

Other examples are even harder to capture

- pala :: palla
  fira :: firra
- atá :: atà
  firé :: firè

???

A new formalism: a modest proposal

Key considerations:

- Can be written by hand
- Can be induced automatically
- Computationally implementable
- **Blazing fast** implementation for induction and application

A new formalism: basic structure

We need a framework with more expressive power:

- Named variables with matching potential
- Segments
- (at some point in the future, maybe) feature structures

For the alternations:

- cant*o* :: cant*aba*
- ca*r*ta :: ca*s*ta
- ca*rt*a :: ca*tr*a

- `[<X1,*>o ⇌ <X1,*>aba]`
- `[<X1,*>r<X2,2>, ⇌ <X1,*>s<X2,2>]`
- `[<X1,*><X2,1><X3,1><X4,1>, ⇌ <X1,*><X3,1><X2,1><X4,1>]`

A new formalism: more patterns

With this system we can cover:

- affixes: prefixes, suffixes and infixes
- metathesis
- reduplication*

We could cover the following if we extended the system with feature structures:

- harmony
- feature alternations

But not:

- morph-positions (Swahili)
- free morph-order (Chintang)

A new formalism: more patterns?

However, we can brute force these problem cases:

- ma$z$ :: ma$s$
- pa$b$ :: pa$p$

Can be covered with independent proportions

- <X,*>z $\rightleftharpoons$ <X,*>s
- <X,*>b $\rightleftharpoons$ <X,*>p

And similarly for harmony and related processes.

A new formalism: generative power?

I have no idea...

It is likely very similar to the generative power of PERL regular expressions.

However, some patterns cannot be captured: multiple free matching variables X*aY* (disallowed by design)

Induction I

Inducing these proportions is straightforward. For a cell pair we do:

- find all optimal alignments between two forms
- non-contrastive material becomes a variable
- contrastive material is left unchanged
- the longest non-contrastive sequence gets a <,*>
- test the coverage of each alignment on all other pairs for the same cell pair
- select the alignment with greatest coverage

Induction II

For example, given: cas*a*n :: *i*cas*o*n

	c	a	s	*a*	n
i	c	a	s	*o*	n

	X1	X1	X1	*a*	X2
i	X1	X1	X1	*o*	X2

1. X1, X1, X1, a, X2↔ i, X1, X1, X1, o, X2
2. `<X1,*>a<X2,1>↔ i<X1,*>o<X2,1>`

Induction III

In the end, we have for each cell pair the following structure:

cell 1	cell 2	proportion
cas*a*	cas*o*	<X1,*>a⇋<X1,*>o
las*a*	las*o*	<X1,*>a⇋<X1,*>o
api	api	<X1,*>⇋<X1,*>
...		

Knowing one cell and the proportion is enough to know the other cell.

Induction IV

Finding non-segmental patterns requires writing look up methods to find those

For example, for metathesis:

- Set a maximum window for metathesis to occur (how many segments can we jump)
- Iterate over an alignment and postulate metathesis as a pattern
- Check if the pattern fits
- Retry

Other types of patterns can be found in a similar way (though I've yet to implement them...)

A new formalism: typological implications

We are making a strong prediction here:

- There are no systems which do: X*aX* $\leftrightharpoons$ X*bX*

As far as I know, this does not exist.

Concluding remarks

What have we learned?

- Formalization is important
- Induction is where we win
  - Most other "formalisms" cannot be induced
  - Induction allows easier exploration of large datasets, or at least assist in the exploration
  - Induction can be made fast and easy
- We need some sort of unification
- It's not clear that we need to capture all patterns found in inflectional morphology, sometimes we can just brute force them into submission

To the demonstration...

Thank you

```
library(tidyverse)
library(analogyR)

ukr <- read_tsv("./ukr.tsv"
                , col_names = c("lexeme", "form", "cell")) %>%
    mutate(cell = cell %>%
                tolower %>%
                str_replace_all(., ";", "_")) %>%
    pivot_wider(names_from = cell, values_from = form) %>%
    select(lexeme:n_dat_sg) %>%
    na.omit()

ukr %>% select(n_acc_sg, n_acc_pl) %>% head
```

```
##   1 абажур        -ø   || абажур        -и
##   2 абажурчик     -ø   || абажурчик     -и
##   3 абаз          -ø   || абаз          -и
```

```
##  4 абазин        -а  || абазин        -ів
##  5 абазинц       -я  || абазинц       -ів
##  6 абазин    -к -у   || абазин    -о -к
```

```
cell_1 <- ukr$n_acc_sg
cell_2 <- ukr$n_acc_pl
```

```
## build analogies between cell1 and cell2
```

```
an_acc_sg_acc_pl <- analogy_build(cell_1, cell_2)
```

```
ukr[1223,] %>% select(n_acc_sg, n_acc_pl)
```

```
## як-і- -ь   як-о- -і
```

```
an_acc_sg_acc_pl[[1223]]
```

```
## [1] "<X1,0> <X3,1> <X2,2> ь  <X1,0> о <X2,2> <X3,1>"
```

```
## [2] "<X1,0> і <X2,2> ь  <X1,0> о <X2,2> і"

##########################################
## check all analogies work:
##########################################

ans_u <- unique(unlist(an_acc_sg_acc_pl))
ans_u %>% length

## we have 62 possible analogies
ans_u

##  [1] "<X1,0>  <X1,0> и"
##  [2] "<X1,0> а  <X1,0> і в"
##  [3] "<X1,0> я  <X1,0> і в"
##  [4] "<X1,0> к у  <X1,0> о к"
##  [5] "<X1,0> у  <X1,0> и"
##  [6] "<X1,0> у  <X1,0>"
```

```
##  [7] "<X1,0> о  <X1,0> а"
##  [8] "<X1,0> ю  <X1,0> ї"
##  [9] "<X1,0> ю  <X1,0> і"
## [10] "<X1,0> ь  <X1,0> і"
## [11] "<X1,0> у  <X1,0> і"

matches <- analogy_fits(cell_1, cell_2, ans_u, .nest = "str

## all analogies work
all(sapply(matches, any))

## check coverage

an_coverage <- as.data.frame(do.call(rbind, matches))

## each column is a patter, each row is a pair the pattern
an_coverage %>% head
```

```
ans_u[1]
## [1] "<X1,0>  <X1,0> и"
cbind(cell_1, cell_2)[1:3,]
## [1,] "абажур"    "абажури"
## [2,] "абажурчик" "абажурчики"
## [3,] "абаз"      "абази"

which(apply(an_coverage, 1, sum)==2)

cbind(cell_1, cell_2)[313,]
##        cell_1        cell_2
## "вар -і- -ь" "вар -о- -і"
ans_u[unlist(an_coverage[313,])]

## [1] "<X1,0> <X3,1> <X2,2> ь  <X1,0> о <X2,2> <X3,1>"
## [2] "<X1,0> і      <X2,2> ь  <X1,0> о <X2,2> і"

## pick best analogies
```

```
colnames(an_coverage) <- ans_u
an_coverage <- colSums(an_coverage, na.rm = TRUE)

an_acc_sg_acc_pl_2 <- sapply(matches, function(mtch) {
        analogy_pick(an_coverage[mtch])
})

ans_u2 <- unique(an_acc_sg_acc_pl_2)

ans_u2 %>% length
## 55

an_acc_sg_acc_pl_2[313]

## [1] "<X1,0> <X3,1> <X2,2> ь  <X1,0> о <X2,2> <X3,1>"
```