НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ

Morphological modelling

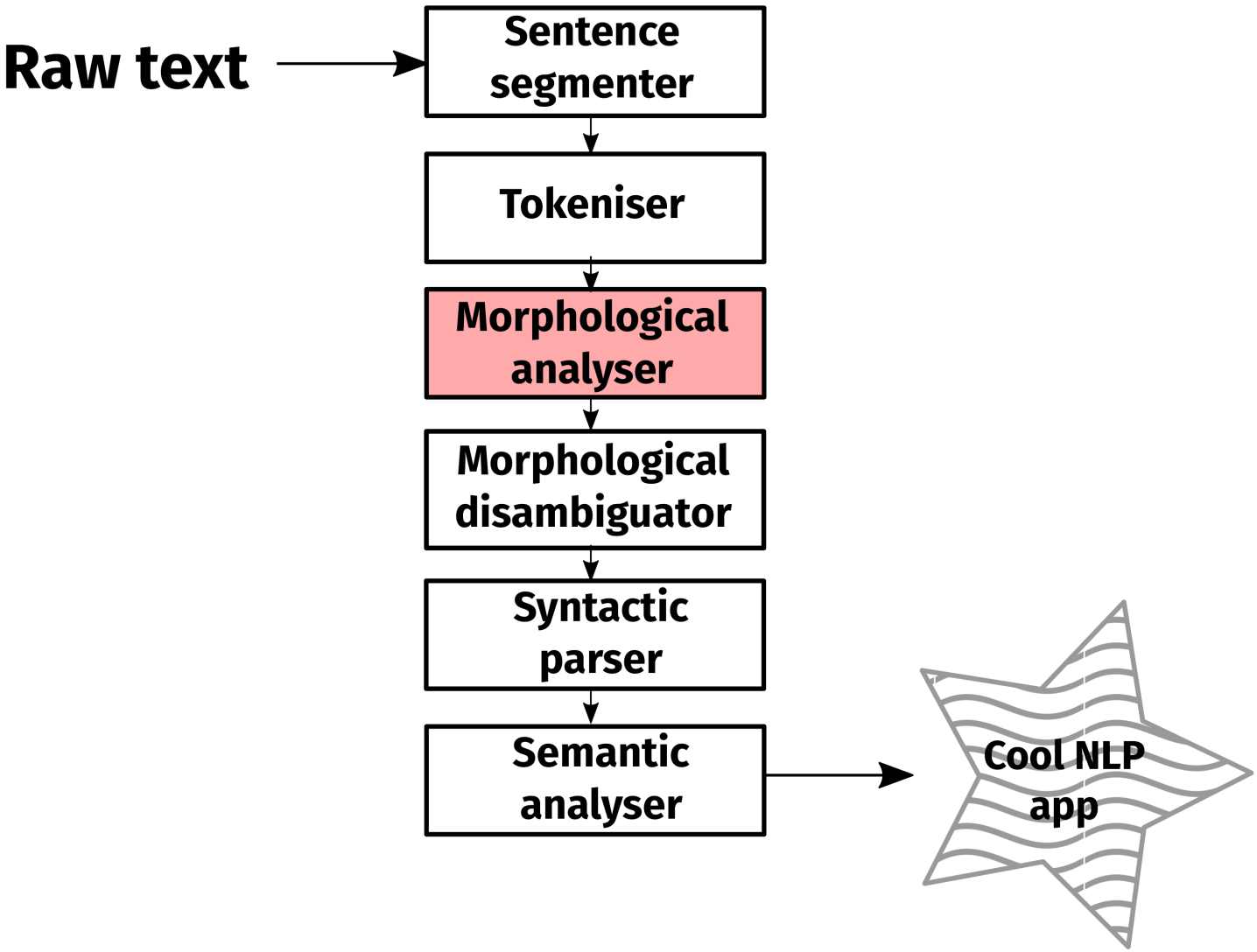
Francis M. Tyers

ftyers@hse.ru
https://www.hse.ru/org/persons/209454856

Национальный исследовательский университет
«Высшая школа экономики» (Москва)

3 ноября 2018 г.

Pipeline


Raw text
Sentence segmenter
Tokeniser
Morphological analyser
Morphological disambiguator
Syntactic parser
Semantic analyser
Cool NLP app

The story so far

В 1942—1945 годах профессором Г. С. Петровым и сотрудниками была разработана серия клеев БФ[1]. Советский учёный-химик Петров знаменит также «контактом Петрова» и работами в области химии и технологии карболита (бакелита, фенолформальдегидных пластмасс)[2].

↓

```
В 1942—1945 годах профессором [[Петров, Григорий Семёнович|Г. С. Петровым]] и сотрудниками была
разработана серия клеев БФ<ref>[http://chem21.info
/page/034120176225149200221127252239157188201019105199/ Справочник по пластическим массам Том 2
(1969) стр.149.]</ref>. Советский учёный-химик Петров знаменит также «контактом Петрова» и
работами в области химии и технологии [[карболит]]а ([[бакелит]]а, фенолформальдегидных
пластмасс)<ref>[http://www.planet-of-people.org/htmls/rus/nadezhdin/plastmassa.htm Надеждин
Н. Я. История науки и техники. Пластмасса<!-- Заголовок добавлен ботом -->]{{Недоступная
ссылка|date=Июль 2018 |bot=InternetArchiveBot }}{{битая ссылка}}</ref>.
```

↓

```
В 1942—1945 годах профессором Г. С. Петровым и сотрудниками была разработана серия клеев БФ.
Советский учёный-химик Петров знаменит также «контактом Петрова» и работами в области химии и
технологии карболита (бакелита, фенолформальдегидных пластмасс).
```

↓

```
В 1942 — 1945 годах профессором Г. С. Петровым и сотрудниками была разработана серия клеев БФ .
Советский учёный-химик Петров знаменит также « контактом Петрова » и работами в области химии и
технологии карболита ( бакелита , фенолформальдегидных пластмасс ) .
```

The story so far

В 1942—1945 годах профессором Г. С. Петровым и сотрудниками была разработана серия клеев БФ[1]. Советский учёный-химик Петров знаменит также «контактом Петрова» и работами в области химии и технологии карболита (бакелита, фенолформальдегидных пластмасс)[2].

↓

```
В 1942—1945 годах профессором [[Петров, Григорий Семёнович|Г. С. Петровым]] и сотрудниками была
разработана серия клеев БФ<ref>[http://chem21.info
/page/034120176225149200221127252239157188201019105199/ Справочник по пластическим массам Том 2
(1969) стр.149.]</ref>. Советский учёный-химик Петров знаменит также «контактом Петрова» и
работами в области химии и технологии [[карболит]]а ([[бакелит]]а, фенолформальдегидных
пластмасс)<ref>[http://www.planet-of-people.org/htmls/rus/nadezhdin/plastmassa.htm Надеждин
Н. Я. История науки и техники. Пластмасса<!-- Заголовок добавлен ботом -->]{{Недоступная
ссылка|date=Июль 2018 |bot=InternetArchiveBot }}{{битая ссылка}}</ref>.
```

↓

```
В 1942—1945 годах профессором Г. С. Петровым и сотрудниками была разработана серия клеев БФ.
Советский учёный-химик Петров знаменит также «контактом Петрова» и работами в области химии и
технологии карболита (бакелита, фенолформальдегидных пластмасс).
```

↓

```
В 1942 — 1945 годах профессором Г. С. Петровым и сотрудниками была разработана серия клеев БФ .
Советский учёный-химик Петров знаменит также « контактом Петрова » и работами в области химии и
технологии карболита ( бакелита , фенолформальдегидных пластмасс ) .
```

Overview

- Morphology: What is it? Why should we care?
- Modelling morphology: With finite-state machines
- Development: Some development tips

Morphology

What is morphology?

Morphology is:

« the branch of linguistics that studies patterns of word formation within and across languages, and attempts to formulate rules that model the knowledge of the speakers of those languages. »

This is a big field, here we are interested in practical models.

Why produce models?

English or Chinese:

- A full form list is a possibility
- Few or no inflectional forms
  - e.g. 5 forms per English verb {see, sees, saw, seen, seeing}

Other languages:

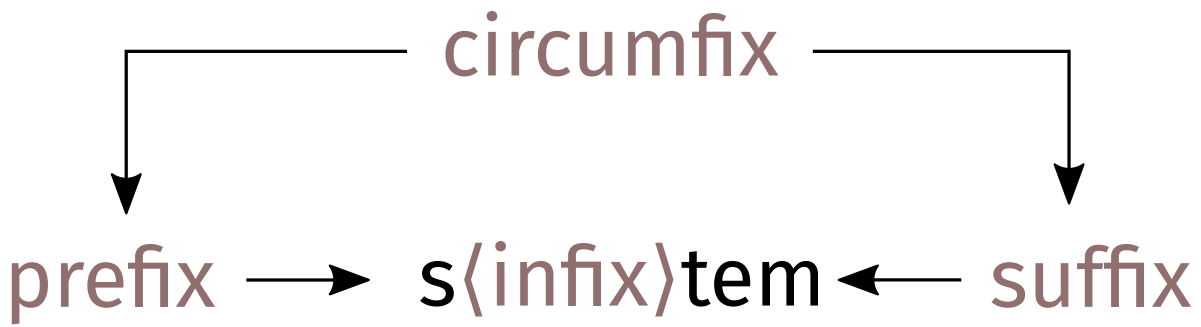
- Difficult or impossible to enumerate all forms
- Very productive inflection and derivation
  - Russian verbs: over 150 forms (maximally)
  - Turkish verbs: thousands of forms

Lexicon

A morphological lexicon consists of entries:

- Lemma: The citation form of a word (cf. headword)
- Stem: The part of a word affixes attach to
- Paradigm: A description of how the word inflects:

Affixes



Add additional meaning or change the meaning of a lexical stem:

- **Suffixes:** *hus* ‘house’ — *huset* ‘the house’
- **Prefixes:** *kjent* ‘known’ — *ukjent* ‘unknown’
- **Infixes:** *opstaan* ‘stand’ — *opgestaan* ‘stood’
- **Circumfixes:** *nagy* ‘big’ — *legnagyobb* ‘biggest’

Morphological processes

- **Inflection:** Inflectional morphemes carry grammatical information, such as number, case, tense, etc., but do not change the word category
- **Derivation:** Derivational morphemes change the basic semantic meaning of a word, and can also change word category.
- **Compounding:** A process where two or more words are joined together to form one, typically of the same category or supertype.
- **Clitics:** Syntactically independent word that functions phonologically as an affix of another word.
- **Incorporation:** Where a nominal (e.g. direct object) or adverbial is included into a verb form.

Inflection

Examples of inflection categories:

- **Case:**
  дом·у ‘house-LOC’, *ev·de* ‘house-LOC’, *talo·ssa* ‘house-INE’
- **Possession:**
  ev·im ‘house-1SG’, *talo·ni* ‘house-1SG’
- **Number:**
  дом·а ‘house-PL, *ev·ler* ‘house-PL’, *talo·t* ‘house-PL’
- **Tense, aspect, mood:**
  говори·ла ‘say-PAST.F, *söyle·di* ‘say-PAST’, *puhu·i* ‘say-PAST’
- **Comparison:**
  больш·е ‘big-COMP’, *пысăк·рах* ‘big-COMP’, *iso·mpi* ‘big-COMP’

In general: Change in meaning is regular.

Derivation

Examples of derivational affixes:

- **Actor**: *diş·çi* /tooth-er/ ‘dentist’
- **State**: *boş·luk* ‘emptiness’, *nycm·oma* ‘emptiness’
- **Diminutive**: *dog·gie*, *kedi·cik* /cat-DIM/ ‘kitten’

Can often be stacked:

- *temizlikçi* /temiz-lik-çi/ clean-ness-er = cleaner
- *поверхностный* /по-верх-ность-ный/ on-surface-ness-ly = superficial

Change in meaning may be irregular, compare:

- *cooker* /cook-er/ ‘machine that cooks’
- *cleaner* /clean-er/ ‘person who cleans’
- *looker* /look-er/ ‘person that looks good’

May be limited to particular stems.

Compounding

New words are formed from morphologically/syntactically independent words:

- This may be indicated in the writing system or not.
  - infrastruktuurontwikkelingsplan, or
  - infrastructure development plan
- tri-noun compounds, but different orthographical treatment

Note: a given compound word may be split different ways, or a given word may appear as a compound, but not be one:

- Freitag = Friday (not “Frei” + “tag” = free day)
- kulturforskeren = the ethnographer, and not
  - kultur+forskeren = “culture researcher”
  - kultur+forske+ren = “culture research clean”

Clitics

Clitics are syntactically separate words that are phonologically conditioned by another unit (word, phrase).

- **Pronominal:**
  - Spanish: *me lo das* me it you.give 'You give it to me'
  - Spanish: *dámelo!* give-me-it 'Give it to me!'
- **Verb forms:**
  - Serbo-Croatian: *govorit ću* vs. *govoriću* 'I will speak'
  - English: *I'm* 'I am', *gonna* 'going to'
- **Other:**
  - Question words (e.g. Finnish *onko?* is-QST? 'Is there?')
  - Tense markers (e.g. Kurdish *-ê*)
- **Consider:**
  - *нравиться* and *собирать, собираться*

Should these be tokenised prior to analysis?

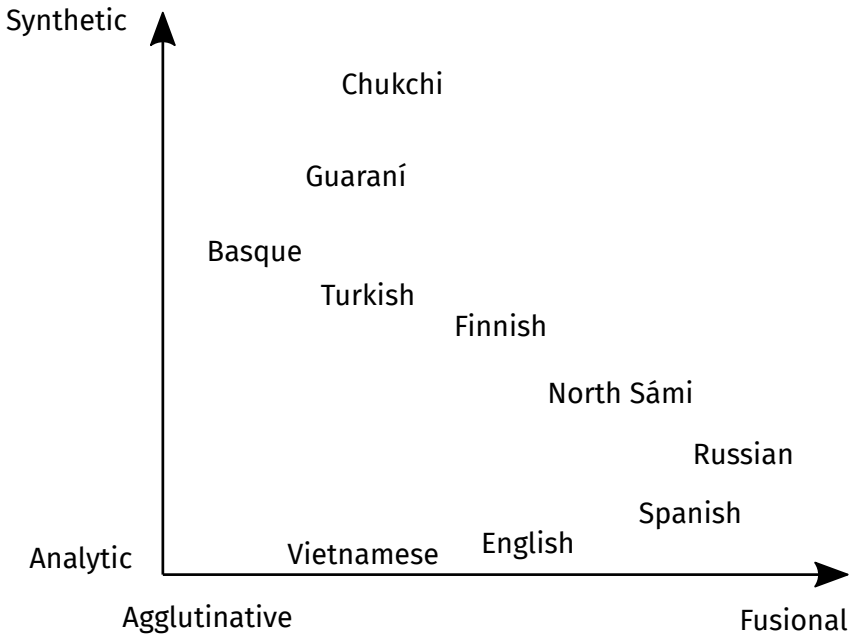
Incorporation

Гаӄорапэнратлэн Сыкваӈаӄай рэмкык
“Cıkwaŋaqaj chased after the reindeer in the other encampment.”

га-ӄора-пэнр-ат-лэн	Сыкваӈаӄай	рэмк-ык
PERF-reindeer-chase-S3SG	Cıkwaŋaqaj	folk-LOC

- Syntactically/pragmatically determined (not lexically!)
- Can be valency changing, e.g.
  - DOBJ + V.TR $\rightarrow$ V.INTR

Morphological typology



- Analytic—Synthetic:
  - Morphemes per word
- Agglutinative—Fusional:
  - Meanings per morpheme / ease of segmentation

Modelling

Analysis and generation

Analysis:

студента $\rightarrow$ {студент<n><m><aa><sg><gen>,
студент<n><m><aa><sg><acc>}

Generation:

студент<n><m><aa><sg><gen> $\rightarrow$ студента

Morphotactics

How morphemes can be combined:

- студентом, играющийся, played, evlerde
- *омстудент, *ющийсяигра, *edplay, *deevler

We'll be using the `lexc` formalism for modelling this.

Morphophonology

The changes that happen when morphemes are combined:

- **Deletion**: работа + ы $\rightarrow$ работы
- **Epenthesis**: fox + s $\rightarrow$ foxes
- **Fleeting vowel**: огонёк + ом $\rightarrow$ огоньком
- **Assimilation**: baş + da $\rightarrow$ başta
- …

Several changes can take place at once:

- өг + ү $\rightarrow$ өө ‘house.3sg’
- Deletion of ‘г’ and assimilation of ‘ү’

We’ll be using the `twol` formalism for modelling this.

Running example

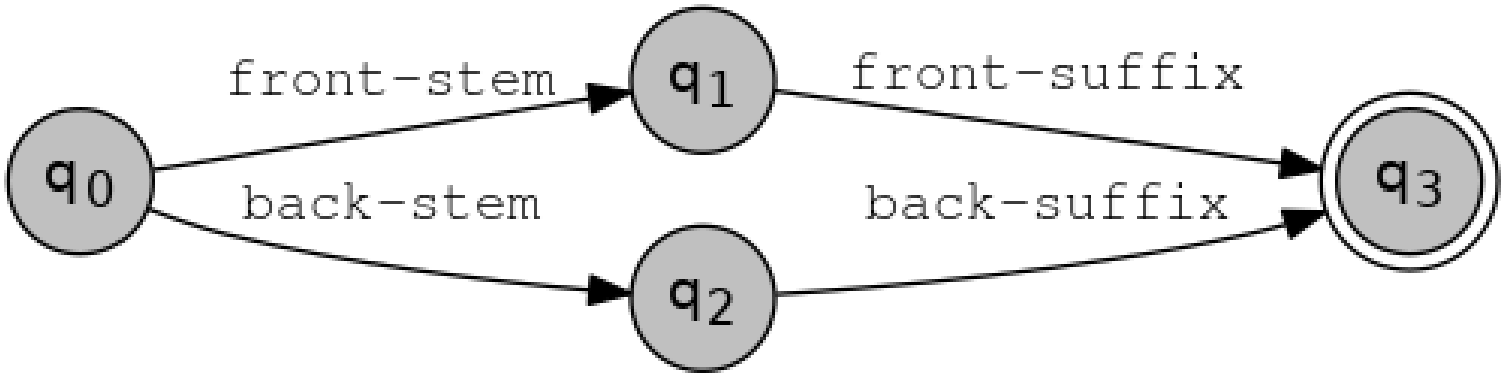
Let's take the Turkish words *ev* 'house', *kız* 'girl':

	Singular	**Plural**
Nominative	ev, kız	ev-ler, kız-lar
Accusative	ev-i, kız-ı	ev-ler-i, kız-lar-ı
Genitive	ev-in, kız-ın	ev-ler-in, kız-lar-ın
Dative	ev-e, kız-a	ev-ler-e, kız-lar-a
Locative	ev-de, kız-da	ev-ler-de, kız-lar-da
Ablative	ev-den, kız-dan	ev-ler-den, kız-lar-dan

Suffixes are different according to front and back vowels.

Finite-state morphology

We can represent these as a finite-state automaton:



Where the labels would mean:

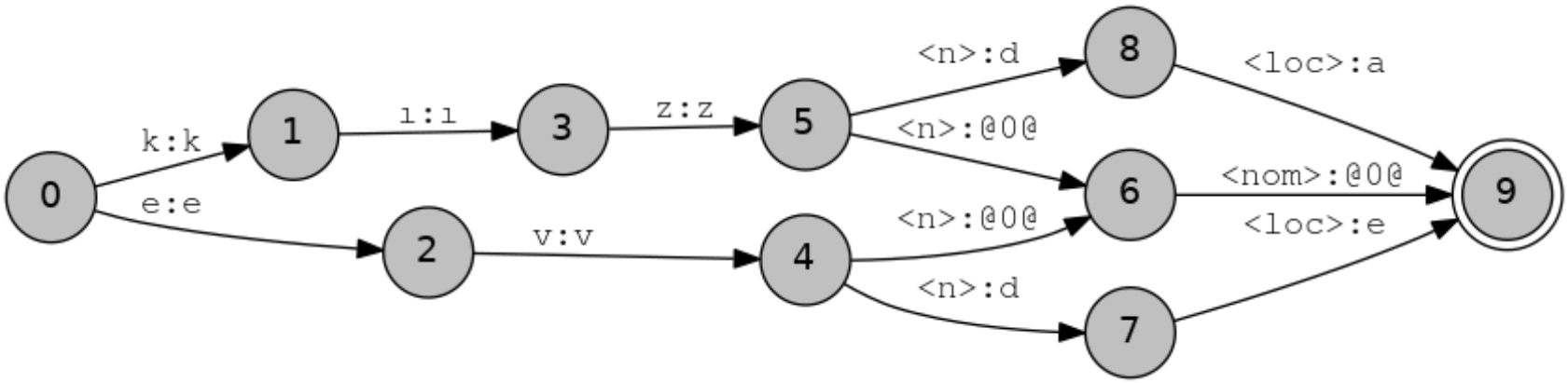
- `front-stem`: the front stems (e.g. *ev*)
- `back-stem`: the back stems (e.g. *kız*)
- `front-suffix`: the front suffixes (e.g. *-de*)
- `back-suffix`: the back suffixes (e.g. *-da*)

Lexicon format: `lexc`

```
Multichar_Symbols

%<n%> %<nom%> %<loc%> %>

LEXICON Root

front-stem ;
back-stem ;

LEXICON front-suffix

%<n%>%<nom%>: # ;
%<n%>%<loc%>:%>de # ;

LEXICON back-suffix

%<n%>%<nom%>: # ;
%<n%>%<loc%>:%>da # ;

LEXICON front-stem

ev:ev front-suffix ; ! "house"

LEXICON back-stem

kız:kız back-suffix ; ! "girl"
```

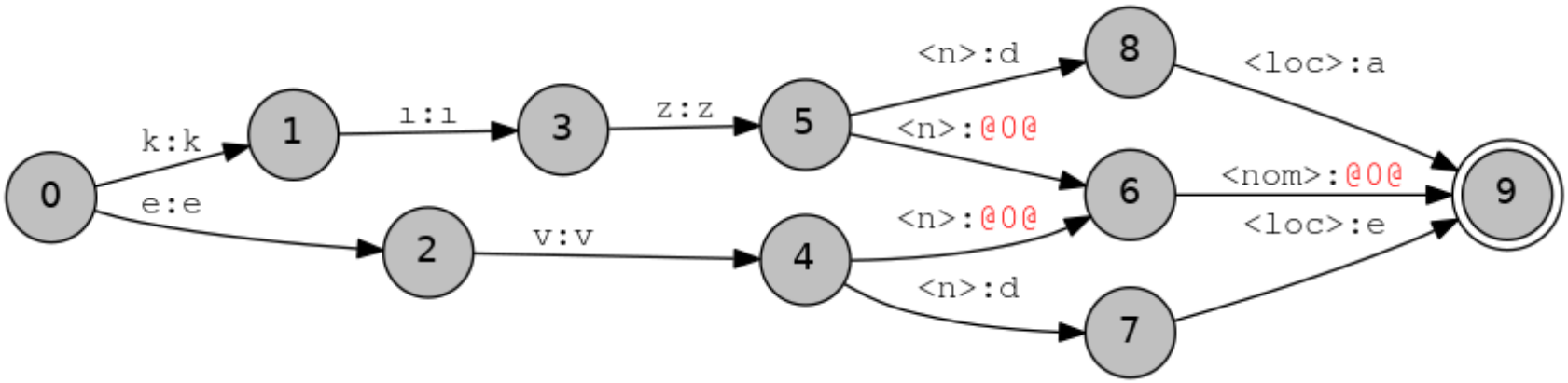
- Tags: Symbols that show grammatical information
- Continuation class: Sets of morphemes
- Next continuation: Shows where to go next
- **#**: End of string
- Comment string: Indicated with **!**
- **%>**: Morpheme boundary

Representing the lexicon



- Q = Set of N states = $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$
- Σ = Input alphabet = $\{\text{a}, \text{d}, \text{e}, \text{k}, \text{ɪ}, \text{v}, \text{z}, \epsilon\}$
- Δ = Output alphabet = $\{\text{e}, \text{k}, \text{ɪ}, \text{v}, \text{z}, <\text{n}>, <\text{nom}>, <\text{loc}>\}$
- $q_0 \in Q$ = A single start state = 0
- $F \subseteq Q$ = A set of final states = $\{9\}$
- $\delta(q, w)$ = A transition function from a state $q \in Q$ and a string $w \in \Sigma^*$ to a set of states in Q

Epsilons

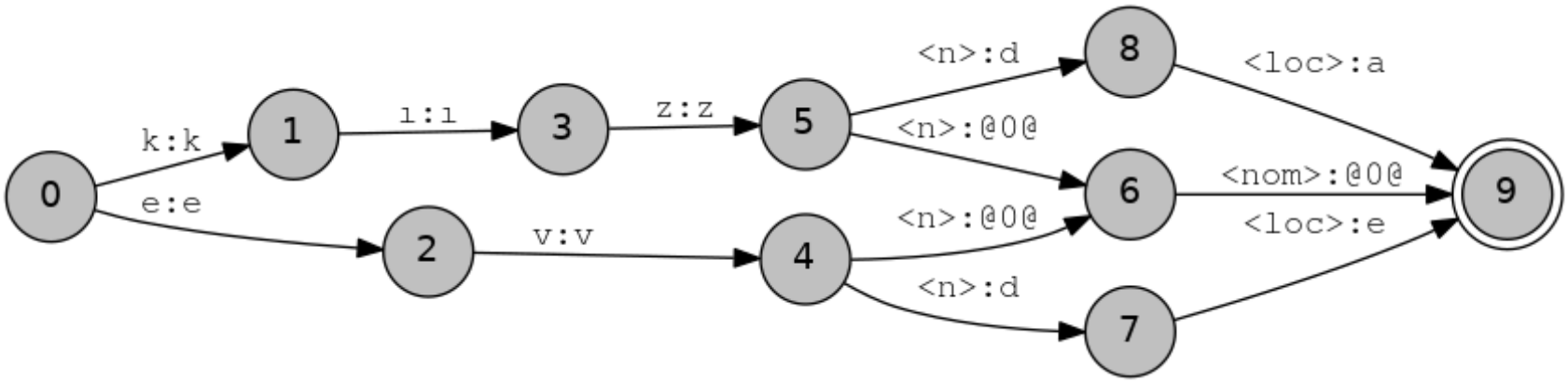


Sometimes we need to input or output a symbol without reading or writing an actual symbol.

- e.g. the $\epsilon \rightarrow$ <n> transition.
- This is commonly encoded as `@0@` and written as ϵ.

Epsilon closure: Finding all states that can be found without reading any input.

Lookup



	Cur. state(s)	In. sym.	Out. state(s)	Out. sym	Out. string(s)
c	0	ϵ	0	–	–
s	0	k	1	k	k
c	1	ϵ	1	–	k
s	1	ı	3	ı	kı
c	3	ϵ	3	–	kı
s	3	z	5	z	kız
c	5	ϵ	6	<n>	kız<n>
c	6	ϵ	9	<nom>	kız<n><nom>

Archiphonemes

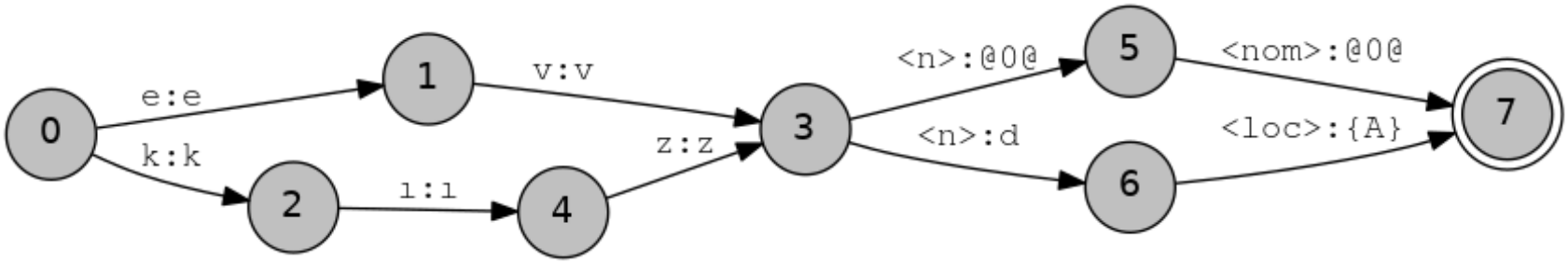
We can simplify the morphotactics by using **archiphonemes**:

- Archiphonemes stand in for underspecified surface symbols
- e.g. underlying %{A%} can be surface *a* or *e*

Example:

```
Multichar_Symbols

%<n%> %<nom%> %<loc%> %{A%}

LEXICON Root

stems ;

LEXICON suffix

%<n%>%<nom%>: # ;
%<n%>%<loc%>:%>d%{A%} # ;

LEXICON stems

ev:ev suffix ; ! "house"
kız:kız suffix ; ! "girl"
```

Archiphonemes/2



- 50% reduction in code length (15 lines $\rightarrow$ 10 lines)
- 20% reduction in number of states (9 states $\rightarrow$ 7 states)

Tapes

It is helpful to think about the transducer as a number of tapes:

Lexical	k	ı	z	0	<n>	<nom>
Morphotactic	k	ı	z	>	d	{A}
Surface	k	ı	z	0	d	a

Objective: Produce a mapping between these tapes

Two-level rules

```
evd{A}:evde
evd{A}:evda
kɪzd{A}:kɪzde
kɪzd{A}:kɪzda
```

[apply rules]

→

```
evd{A}:evde
kɪzd{A}:kɪzda
```

- First expand all possible forms on the morphotactic tape
- Rules are constraints on possible symbol pairs
- Each rule is an automaton which accepts or rejects a string

Schema of a rule file

```
Alphabet
a b c d e f g h i j k l m n o p q r s t u v
w x y z ü ö ş ç ı %{A%}:a %{A%}:e ;

Sets
Back = a ı o u ;
Cns = b c d f g h j k l m n p q r s t v w x y z ş ç ;

Rules
```

Three main sections:

- **Alphabet**: Valid symbol **pairs**, n.b. `a` = `a` : `a`, etc.
- **Sets**: Groups of symbols to be used in rules
- **Rules**: Constraints

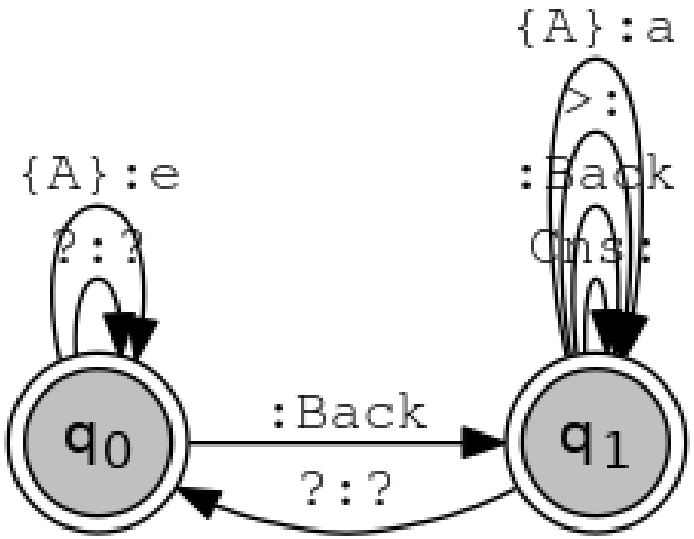
Rule example: Vowel harmony

```
"Vowel harmony"
%{A%}:a <=> :Back [ Cns: | :Back | %>: ]* _ ;
```

- Symbol pair: The symbol pair to constrain
- Rule operator: The type of constraint
- Rule context: The context where the rule should apply
- Rule centre: Where the symbol pair is found in the context

Rule example: Vowel harmony

```
"Vowel harmony"
%{A%}:a <=> :Back [ Cns: | :Back | %>: ]* _ ;
```



k	ɪ	z	d	{A}
k	ɪ	z	d	a
k	ɪ	z	d	**{A}**
k	ɪ	z	d	**e**
				↑
				FAIL

Rule operators

	Positive Reading	*Negative Reading*
`a:b <=> l _ r ;`	1. If the symbol pair a:b appears, it must be in the context l _ r. 2. If lexical a appears in the context l _ r, then it must be be realized on the surface as b.	1. If the symbol pair a:b appears outside the context l _ r, FAIL. 2. If lexical a appears in the context l _ r and is realized as anything other than b, FAIL.
`a:b => l _ r ;`	If the symbol pair a:b appears, it must be in the context l _ r.	If the symbol pair a:b appears outside the context l _ r, FAIL.
`a:b <= l _ r ;`	If lexical a appears in the context l _ r, it must be realized on the surface as b.	If lexical a appears in the context l _ r and is realized as anything other than b, FAIL.
`a:b /<= l _ r ;`	Lexical a is never realized as b in the context l _ r .	If lexical a is realized as b in the context l _ r, FAIL.

Table 1.1: **twolc** Rule Operator Semantics

From `twolc.pdf` page 22

Running example/2

Sometimes several rules can apply to the same form:

	Singular	**Plural**
Nominative	ev, kız, baş	evler, kızlar, başlar
Accusative	evi, kızı, başı	evleri, kızları, başları
Genitive	evin, kızın, başın	evlerin, kızların, başların
Dative	eve, kıza, başa	evlere, kızlara, başlara
Locative	evde, kızda, başta	evlerde, kızlarda, başlarda
Ablative	evden, kızdan, baştan	evlerden, kızlardan, başlardan

The suffix *-da* can be *-ta*/*-te*, e.g. *başta* 'head-LOC' not **başda*.

- This calls for another archiphoneme! `%{D%}` → `{d, t}`

```
Multichar_Symbols

%<n%> %<nom%> %<loc%> %{A%} %{D%}

LEXICON Root

stems ;

LEXICON suffix

%<n%>%<nom%>: # ;
%<n%>%<loc%>:%>%{D%}%{A%} # ;

LEXICON stems

ev:ev suffix ; ! “house”
kız:kız suffix ; ! “girl”
baş:baş suffix ; ! “head”
```

Rule application process

Input

```
ev>{D}{A}
kız>{D}{A}
baş>{D}{A}
```

Expand

```
ev>{D}{A}:evda
ev>{D}{A}:evde
ev>{D}{A}:evta
ev>{D}{A}:evte
kız>{D}{A}:kızda
kız>{D}{A}:kızde
kız>{D}{A}:kızta
kız>{D}{A}:kızte
baş>{D}{A}:başda
baş>{D}{A}:başde
baş>{D}{A}:başta
baş>{D}{A}:başte
```

Apply rules

```
ev>{D}{A}:evde
kız>{D}{A}:kızda
baş>{D}{A}:başta
```

Rule application/1

```
"Vowel harmony"
%{A%}:a <=> :Back [ Cns: | :Back | %>: ]* _ ;
```

```
"Voicing assimilation"
%{D%}:t <=> :Unvoiced %>: _ ;
```

- Rules are applied in parallel
- Every pair must be accepted by all rules

Rule application/2

Modelled as the **composition** of the lexicon with each rule in turn:

Vowel harmony:

		Accepted strings
baş%>%{D%}%{A%}	başda	başda
baş%>%{D%}%{A%}	başta	başta
baş%>%{D%}%{A%}	başde	başde
baş%>%{D%}%{A%}	başte	başte

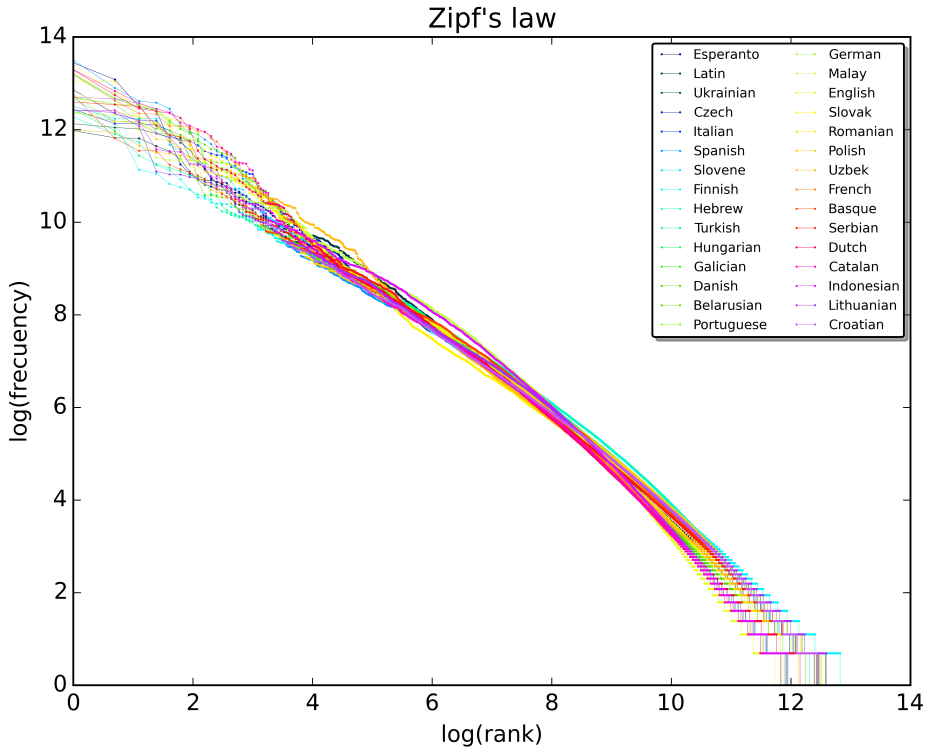
Voicing assimilation:

		Accepted strings
baş%>%{D%}%{A%}	başda	başda
baş%>%{D%}%{A%}	başta	başta
baş%>%{D%}%{A%}	başde	başde
baş%>%{D%}%{A%}	başte	başte

And then **intersection:** $\{\text{başda}, \text{başta}\} \cup \{\text{başte}, \text{başta}\} = \{\text{başta}\}$

Development

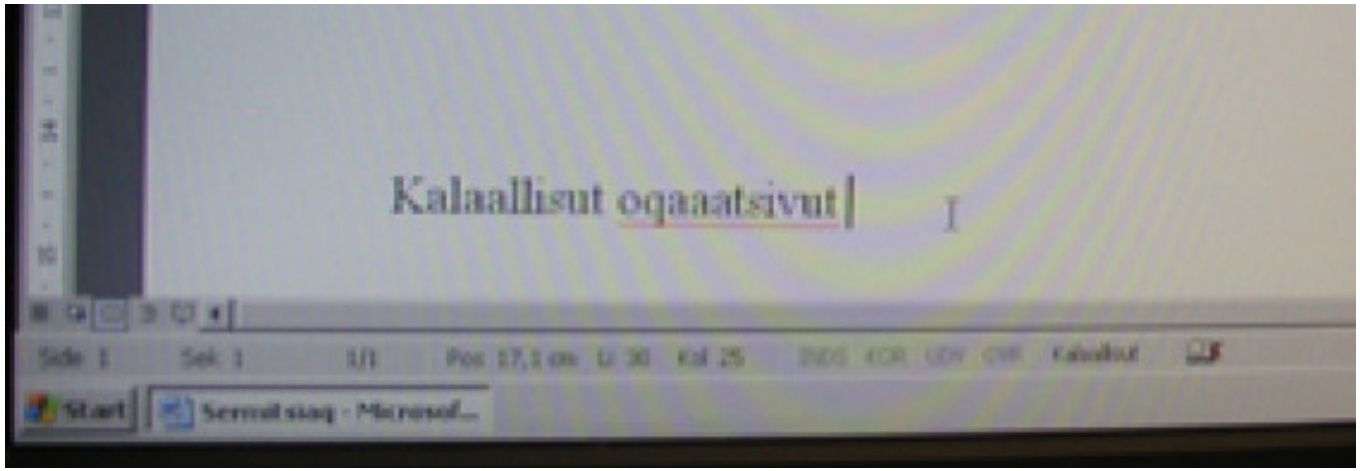
Development guidelines



Take **frequency** into account, in adding:

- Stems
- Morphemes
- Phonological rules

Applications



- **Spellcheckers:** For morphologically-rich languages that have little data, FSTs are the only choice.
- **Online dictionaries:** For languages where it is non-trivial to determine the headword from a surface form, an FST can be a real aid
  - for learners and for newly literate speakers
- **Improve parsing:** For languages with limited data for training a parser, an FST can significantly improve performance.

What we have not covered

- **Templatic morphology:**
  - Semitic languages like Maltese, Hebrew and Arabic use templates to form surface forms, e.g. Maltese *k-t-b* could be *ktieb* 'book' or *kotba* 'books'
  - The FSMBook[1] has examples of how to treat these
- **Machine learning approaches:**
  - Recent advances in morphological generation (SIGMORPHON)[2]
  - Morphological analysis way behind
- **Rewrite rules:**
  - Some prefer to write phonological rules as a cascade of rules
  - Computationally equivalent
  - See FSMBook for further details
- **Weighting:**
  - Refer to the practical

[1] Beesley and Karttunen (2003) *Finite-State Morphology* (Chicago: CLSI)

[2] https://sigmorphon.github.io/sharedtasks/

Practical

Go through the following practical:

`https://ftyers.github.io/2017-КЛ_МКЛ/hfst.html`

This will take you through all of the main steps to build a transducer.