Computational morphology

Summer School
in
Computational Linguistics and Represented Knowledge
Sozopol, Bulgaria
August 25 - September 8, 2000

Kiril Iv. Simov & Guergana D. Popova

What is Morphology

- How do we define morphology?

Knowledge of word structure

Compare;

`disestablishmentarianism`

vs.

`ismdisarianestablishantiment` or

`mentestablishdisantiismarian`

- How do we define word?
  - **orthographic word** - string of letters with a blank space on either side
  - **LEXEME** - an abstract representation of a word which can be realized by different word forms

Lexeme:	`LOVE`
Word forms:	*love*
	loves
	loving
	loved

  - **grammatical word**

```
He scored a goal.
He has never scored a goal.
```

What is a morpheme?

- cannot be decomposed any further
- contributes "meaning"
- it is an abstract element of analysis

Morph - segment of a word form which represents a particular morpheme

Allomorph

cats, roots	`/s/`
horses, courses	`/iz/`
tables, doors	`/z/`

index - *indices*

house - *houses*

knife - *knives*

Schwachheit -en `weaknesses`

Aüssigkeit -en `fluids`

Allomorphy can be conditioned by phonology, by the presence of other morphemes or it can be an idiosyncratic property of words

Suppletion /partial and total/

go - *went*

good - *better* - *best*, *good* - *well*

despise - *contempt*

We also distinguish:

- **Free mopheme** - a morpheme which can stand alone as a complete word e.g. *book, green, go*
- **Bound morpheme** - a morpheme which must be accompanied in all its occurrences by one or more additional morphs to produce a well-formed word e.g. *re-* (rewrite), *-er* (writer), *-ing* (writing)
- **Root** - the simplest possible form of a lexical morpheme upon which all other bound or free forms involving that morpheme are based
- **Base** - a morph, variously consisting of a root, a stem, or a word, which serves upon the addition of a single further morpheme, as the immediate source of some particular formation: e.g. *happy* is the base for both *unhappy* and *happily*, *unhappy* is the base for *unhappily* and *unhappiness*
- **Stem** - we need to distinguish stem mostly when dealing with inflectional morphology. It is that part of the word which remains once it has been stripped of all inflectional affixes

- **Affixes** - bound morphemes which can occur only attached to a word or a stem
  - **suffix** - attached after the base e.g. *cat-s. writ-er, happi-ness*
  - **prefix** - attached before the base e.g. *un-happy, re-tell*
  - **infix** - inserted in a morph
  - **circumfix** - two affixes attached simultaneously with a single meaning

Bloomfield's classification

How do languages encode information in their morphology?

- **Isolating languages** - without bound forms

  Mandarin Chinese

  gǒu bú ài chi qingcài

  `(dog not like eat vegetable)`

- **Agglutinative languages** - "beads on a string"

  Turkish

  çöp+lük+ler+imiz+de+ki+ler+den+mi+y+di

  `(garbage + AFF+PL+1P/PL+LOC+REL+PL+ABL+INT+AUX+PAST)`

  `was it from those that were in our garbage cans?`

- **Polysynthetic languages** - encode morphologically elements that are sepatrate words in other languages

  Central Alaskan Yupik:

  qayá:liyú:lú:ní

  `he was excellent(-yu-) at making (-li-) kayaks (qaya)`

- **Inflecting languages** - merge features into a single bound form

  Bulgarian

  doid-oh

  `came - 1P/SG/PAST`

Morphological Operation

Two main types of morphological operation

- **Inflection**
  - does not change the syntactic category of the word
  - very often is required by the syntactic context
  - changes the grammatical characteristics of the word
  - more or less fully productive
- **Derivation**
  - most often is not required by the suntactic context
  - typically changes the syntactic category of a word
  - not fully productive

Morphological class - more or less arbitrary grouping of words which are associated with different sets of inflections

Paradigm - the set of all inflected forms of an individual word.

Morphemes as things

Affixation

Infix - an affix which is placed inside another morpheme

Tagalog

sulat (write - single morph)

s-um-ulat (to write - subject focus)

s-in-ulat (to write - direct object focus)

Circumfix - a prefix and a suffix attach to a base simultaneously to express a single meaning or a category conjointly

German past participle

ge-wander-t (wandered)

Clitics - (1) cannot exist independently, hence they are bound morphemes; (2) they attach themselves to fully inflected words

Bulgarian.

Dadoh mu go (I gave it him - I gave it to him)

Compounding

Noun+noun = noun : *teapot, vodopad* (waterfall)

Adj.+noun = noun : *blackbird*

noun+adj. = adj. : *Cobalt-blue*

verb+noun = noun : (rare) *swearword*

verb+object : *postalettere* (postman Ital.) "carries letters"

whole phrase : *cessez-le-feu* (ceasefire French) "stop the fire"

Morphemes as rules

Stress

to contrast - a contrast

to increase - an increase

to import - import

Ablaut - change in the vowels of the root

sing-sang-sung

tooth - teeth

mouse - mice

Conversion

to cut - a cut

to run - a run

to ring - a ring

a hand - to hand

an orbit - to orbit

The Lexicon

What do we find in traditional dictionaries?

- spelling
- pronunciation
- meaning
- morphological properties
- syntactic properties
- idiosyncratic information

American structuralists

- the lexicon should contain only idiosyncratic information
- grammar - a set of word formation rules
- lexicon = list of morphemes

Alternative approach

- lexicon = list of complete words
- potential words vs. actual words
- lexicon = permanent lexicon

or

- lexicon - list of morphemes + list of words formed by unproductive morphological processes

What are the applications of morphological studies?

- in a parsing system - to identify the part of speech
- in a generating system - to select a word appropriate for the context
- to find the word boundaries in languages like Chinese
- for dictionary look-up - to find all words morphologically related to our original query
- for document retrieval - to find the whole paradigm when searching for a key word
- for creating or expanding dictionaries
- text-to-speech systems - morphological analysis to predict correct pronunciation, e.g. boathouse
- to minimize the lexicon when the meaning of a morphologically complex word is "computable"

Computational Morphology

The **goal** of computational morphology is to represent morphological knowledge in a linguistically adequate way so as to support the following tasks:

1. **Analysis**: to give information about the internal structure of word forms, the grammatical information carried by the elements of this structure and point to the lexeme to which a word-form belongs
2. **Generation**: given some lexeme and relevant grammatical information generate a word form which carries this grammatical information
3. **Linguistic investigation**: support further acquisition of linguistic knowledge

Morphological information can be described as consisting of two parts:

1. Information about the phonemes (or morphemes) of the language
2. Grammatical or lexico-semantic information that morphemes encode

Various computational models differ in the amount of information they represent and the way they do it

Two-Level Morphology

Uses two levels each with its own alphabet

- On the **surface level** words are represented as they are written (or pronounced) in the natural language (plus the 0)
- The **lexical level** includes also some special symbols (diacritics) which give necessary phonological information but are not represented as graphemes (phonemes).

Words on the lexical level are called lexical words, on the surface level they are called surface variants of lexical words.

A lexical word and its surface variants are required to be of equal length.

The correspondence between a lexical word and its surface variant is given by a set of rules that map each of the symbols in the first to the corresponding symbol in the second.

Every single character from one level maps to one character from the other level. By default a letter common to both alphabets maps to itself, the word and the morph boundaries map to the zero symbol.

Two-Level Rules

Formally, each rule has the following form:

$$LS : SS \circ LCont - RCont$$

LS – symbol on the lexical level

SS – corresponding symbol on the surface level

$\circ$ is the rule operator ($\Leftarrow, \Rightarrow, \Leftrightarrow, /\Leftarrow$)

$LCont$ and $RCont$ are regular expressions over pairs of symbols which define when the correspondence is valid

- **Context restriction rule ($\Leftarrow$).** The lexical symbol LS may be realized as surface symbol SS only in the given context and nowhere else.
- **Surface coercion rule ($\Rightarrow$).** The lexical symbol LS must be realised as the surface symbol SS in the given context.
- **Composite rule ($\Leftrightarrow$).** A combination of the previous two; the lexical symbol LS must correspond to the surface symbol SS in the given context, and this correspondence is licit only in this context.
- **Exclusion rule ($/\Leftarrow$).** The lexical symbol LS may not be realized as the surface symbol SS in the context $LCont\_RCont$.

Context Descriptions

In the description of contexts the following syntactic means are used:

- Each pair of a lexical and a surface symbol is a context description;
- Context descriptions in **curly brackets** { } are regarded as alternatives;
- Context descriptions in **square brackets** `[ ]` denote concatenation;
- Some sets of pairs are abbreviated by a **name**, e.g. V stands for the set of vowels (`a:a, e:e ...`). Such a name in a context description denotes occurence of exactly one of the pairs in the set;
- If such a name is immediately followed by the + sign this means that **one or more** pairs from the set can occur in this position. When they are more than one they are connected with [ ];
- If the name is immediately followed by `*` this means that **zero or more** pairs from the set can be in this position. When they are more than one they are connected with [ ];
- If in a pair there is nothing to the left or to the right of the colon then any symbol can occupy this position.

Examples of Context Descriptions

- `{ s:s x:x z:z }` denotes contexts in which only one of the pairs `s:s, x:x, z:z` appears.
- `[ s:s h:h ]` denotes contexts in which on the lexical and on the surface level the string `sh` appears.
- `[ s:s h:g ]` denotes contexts in which on the lexical level `sh` occurs and on the surface level the string `sg` appears.
- `V` denotes contexts such that one of the pairs `a:a, e:e, i:i, o:o, u:u` appears.
- `a:` denotes contexts such that on the lexical level `a` occurs and on surface level an arbitrary symbol appears.
- `:a` denotes contexts such that on the surface level `a` occurs and on lexical level an arbitrary symbol appears.
- `[V+ C]` denotes contexts such that one or more vowels is followed by a consonant.
- `{s:s x:x z:z [ { s:s c:c } h:h ]}` denotes contexts in which `s, x, z, sh` or `ch` occurs.
- `+:0` denotes contexts in which the morpheme boundary on the lexical level corresponds to the zero symbol on the surface level.

Examples of Rules

`wife - wives`

Lexical level: `wife+s$`

Surface level: `wive0s0`

$$\texttt{f:v} \Rightarrow \texttt{V \_ [e:e +:0 s:s]}$$

`dish - dishes`

Lexical level: `dish+s$`

Surface level: `dishes0`

$$\texttt{+:e} \Leftrightarrow \{ \texttt{ s:s x:x z:z [} \{\texttt{s:s c:c}\} \texttt{ h:h]} \} \texttt{ \_ s:s}$$

`love - loved`

Lexical level: `love+ed$`

Surface level: `lov00ed0`

$$\texttt{e:0} \Leftrightarrow \texttt{ C \_ [+:0 e:e d:d]}$$

More Examples

`potato - potatoes`:

Lexical level: `potato+s$`

Surface level: `potatoes0`

$$\texttt{+:e} \Leftarrow \texttt{o:o \_ s:s}$$

Exceptions: `banjo - banjos, banjoes` and `piano - pianos`.

Lexical level: `piano#+s$`

Surface level: `piano00s0`

$$\texttt{+:0} \Leftarrow \texttt{[o:o \#:0] \_ s:s}$$

Lexical level: `banjo@+s$`

Surface level: `banjo0es0`

`banjo00s0`

$$\texttt{+:e} \Leftarrow \texttt{[o:o @:0] \_ s:s}$$

$$\texttt{+:0} \Leftarrow \texttt{[o:o @:0] \_ s:s}$$

Bulgarian Examples

`kotel - kotelat`, `učitel - učitel′at`

Lexical level: `učiteL+at$`
Surface level: `učitel0′at0`

Lexical level: `kotel+at$`
Surface level: `kotel0at0`

`L:l` ⇔ _

`a:′a` ⇔ `[L:l +:0]` _ `[t:t $:0]`

`gastalak - gastalaci`

Lexical level: `gastalaK+i$`
Surface level: `gastalac0i0`

`K:k`/⇐ _ `[+:0 i:i]`

`K:c` ⇔ _ `[+:0 i:i]`

Paradigmatic Morphology

Paradigmatic Morphology is based on the traditional notion of Word and Paradigm. It provides means for declarative statements of morphological relationships and lexical rules. Constraints on the orthography are expressed by string equations. Grammatical information is represented via paradigms ordered by subsumption.

String Equations and String Unification

A **string** S is a sequence of elements drawn from a finite alphabet A combined by the associative operator $+$, representing the concatenation of strings.

A **string specification** or **string form** is a sequence possibly containing variables drawn from a set of variables V, disjoint from A. The operator $+$ can be omitted.

`Examples`: (lower case letters are from the alphabet A and upper case letters are variables).

`w+a+l+k+s`	`walks`
`s+ö+r`	`sör`
`A`	`A`
`W+s`	`Ws`
`k+V+t+V+b`	`kVtVb`

String Equations and Unification

String specifications are partial descriptions of strings.

An **equation** of two string specifications represents the fact that they denote the same string. Some equations follow with an **assigment** for the variables that satisfy the equations:

```
walks = Ws         W/walk
sörAk = XYZYW      A/ö, Y/ö, X/s, Z/r, W/k
kVtWb = CiDaE      V/i, W/a, C/k, D/t, E/b
```

String unification ($\sqcup$) is an operation on two string specifications which determines the most specific variable assignment for the variables in the specifications that converts both sides of an equation in the same object. In the general case, string unification is **undecidable**.

Paradigmatic morphology uses a **special class** of equations, such that repeated variables are only permitted on one side of the equation. For them the unification problem is **decidable**.

Subsumption Relation

String specifications are partially ordered by the **subsumption relation**.

A string specification S **subsumes** another S' ($S \sqsubseteq S'$) if all ground instances (instances without variables) of S' are also instances of S. Equivalently, $S \sqsubseteq S'$ if the unification of S and S' is S' ($S' \sqcup S = S'$). S and S' are **incompatible** if and only if $S \sqcup S'$ is undefined.

Lexical Items

Each **lexical item** $S : P$ associates a string S (no variables in S) and a set of grammatical properties P. The grammatical properties are atomic.

A **lexical specification** is a pair $\sigma : \phi$ where σ is a string specification and ϕ is a set of properties.

A lexical specification $\sigma : \phi$ **subsumes** another $\sigma' : \phi'$ if and only if $\sigma \sqsubseteq \sigma'$ and $\phi \subseteq \phi'$.

The **lexicon** consists of a finite set of lexical items.

Lexical Rules

Each **lexical rule** is a triple

$$< Name, IS : IP, OS : OP >$$

where

IS and OS are string specifications

IP and OP are sets of grammatical properties

$Name$ is a name

Each **grammatical property** is an atomic symbol.

Each lexical rule represents a mapping between the set of **"input"** properties IP and the set of **"output"** properties OP.

The **interpretation** of a rule with respect to grammatical properties is as follows:

Given a lexical item $S : P$

and a lexical rule $< LR, IS : IP, OS : OP >$,

LR relates P to another set of properties P' (its "output") in the following way:

$$P' = (P \setminus IP) \cup OP$$

where $OP \subseteq P'$ and $IP \subseteq P$.

The relationship between the string specifications IS and OS is mediated by a **paradigm**.

Paradigms

Each **paradigm** is a quadruple

$$< Name, \sigma : \phi, [LR_1 \ldots LR_n], [S_1 \ldots S_n] >, n >= 1$$

which relates string forms σ and S_i via the lexical rule LR_i under condition ϕ
where any variables in S_i also occur within σ
S_i is a *derived string form*
$Name$ is the (unique) name of the paradigm
$\sigma : \phi$ is the lexical specification of the paradigm

Paradigm Interpretation

If a paradigm

$$< Name, \sigma : \phi, [LR_1 \ldots LR_i \ldots LR_n], [S_1 \ldots S_i \ldots S_n] >$$

is applicable, then
lexical items $S : P$ and $S_i : P'$ are related by lexical rule $< LR_i, IS : IP, OS : OP >$ with $P' = (P \setminus IP) \cup OP$.

Paradigm Applicability

For a paradigm

$$< Name, \sigma : \phi, [LR_1 \ldots LR_i \ldots LR_n], [S_1 \ldots S_i \ldots S_n] >$$

to be applicable to a lexical item $S : P$, two conditions must hold:

1. $\sigma \sqsubseteq S$ and $\phi \subseteq P$
2. There is no paradigm
$< Name, \sigma' : \phi', [LR'_1 \ldots LR'_i \ldots LR'_n], [S'_1 \ldots S'_i \ldots S'_n] >$
such that $\sigma \sqsubseteq \sigma'$, $\phi \subseteq \phi'$, $\sigma' \sqsubseteq S$ and $\phi' \subseteq P$.

The first condition requires that the lexical specification of the paradigm subsume the lexical item.

The second one requires that there be no paradigm whose lexical specification is more specific and which is also applicable to the lexical item.

Given a lexicon containing a finite number of lexical items (called also lexical entries), paradigms and lexical rules, the set of lexical items is defined as the closure of the lexicon under the application of lexical rules mediated by the paradigms.

Example

```
paradigm(verb, Verb:[verb,base],
   [base 3sg    non3sg past_participle (1)
   [Verb Verb+s Verb   Verb+ed         (2)
           (1) past    passive progressive]
           (2) Verb+ed Verb+ed Verb+ing] )

paradigm(verb_age, age:[verb,base],
   [progressive]
   [ageing] )

paradigm(verb_strong,
       S+in+C:[verb,base,strong,C={g,k}],
   [past_participle past     passive]
   [S+un+C          S+an+C   S+un+C])

paradigm(verb_bring, bring:[verb,base,strong]
   [past_participle past     passive]
   [brought         brought  brought])

lex_rule(3sg,
   [verb, base] -> [vebr, finite, 3sg])

lex_rule(non3sg,
   [verb, base] -> [vebr, finite, non3sg])
```

The Morpho-Assistant Model

Elena Paskaleva, Mariana Damova, Milena Slavtcheva, Tania Avgustinova, Kiril Iv. Simov

A system for inflectional morphology which carries out: **Analysis** and **Synthesis**

Paradigm Representation

List of grammatical information for each member of the paradigm:

$$Paradigm(noun\_masc) \Rightarrow$$

$$\left\{ \begin{array}{c} \left[Paradigm\ Gram.\ Char.: \left[\begin{array}{ll} Number: & Singular \\ Case: & Nom \end{array} \right] \right] \\ \vdots \\ \left[Paradigm\ Gram.\ Char.: \left[\begin{array}{ll} Number: & Plural \\ Case: & Acc \end{array} \right] \right] \end{array} \right\}$$

Morphological Classes

Each morphological class is a pair of two descriptions: **inflective type** which determines the endings for the members of the paradigm; **alternation type** which determines the alternations in the stem for some members of the paradigm.

Inflectional Types

$$
\mathit{Inflective\ Type(ft\_Grund)} \Rightarrow
\left\{
\begin{array}{l}
\left[\begin{array}{ll}
\mathit{Ending:} & \emptyset \\
\mathit{Paradigm\ Gram.\ Char.:} & \left[\begin{array}{ll} \mathit{Number:} & \mathit{Singular} \\ \mathit{Case:} & \mathit{Nom} \end{array}\right]
\end{array}\right] \\
\vdots \\
\left[\begin{array}{ll}
\mathit{Ending:} & \mathit{en} \\
\mathit{Paradigm\ Gram.\ Char.:} & \left[\begin{array}{ll} \mathit{Number:} & \mathit{Plural} \\ \mathit{Case:} & \mathit{Acc} \end{array}\right]
\end{array}\right]
\end{array}
\right\}
$$

Alternational Types

$$
\mathit{Alternation\ Type(alt\_Grund)} \Rightarrow
\left\{
\begin{array}{l}
\left[\begin{array}{ll}
\mathit{Alt:} & < u \rightarrow \ddot{u}, \mathit{position:\ 3} > \\
\mathit{Paradigm\ Gram.\ Char.:} & [\mathit{Number:\ Plural}]
\end{array}\right] \\
\left[\mathit{Paradigm\ Gram.\ Char.:}\ \neg[\mathit{Number:\ Plural}]\right]
\end{array}
\right\}
$$

Lexeme Information

$$
\left[\begin{array}{l}
\mathit{Lexeme:\ GRUND} \\
\mathit{Gram.\ Char.:} \left[\begin{array}{ll}
\mathit{Part\ of\ Speech:} & \mathit{Noun} \\
\mathit{Gender:} & \mathit{Masculine} \\
\mathit{Add.\ information:} & < \ldots >
\end{array}\right] \\
\mathit{Stem:\ Grund} \\
\mathit{Paradigm:\ noun\_masc} \\
\mathit{Inflective\ Type:\ ft\_Grund} \\
\mathit{Alternation\ Type:\ alt\_Grund}
\end{array}\right]
$$

The RPPP Model

Dimitar G. Popov and Kiril Iv. Simov

Primary goal: - adequate representation of Bulgarian inflectional morphology

Analysis and synthesis are secondary goals

Paradigmatic approach.

Very simple inflectional classes common for most of the lexemes.

Two-level-like rules connecting strings with arbitrary lengths with grammatical information as context.