



Computational Morphology



Author: Harald Trost

Abstract

Computational morphology deals with the processing of words and word forms, in both their graphemic, i.e., written form, and their phonemic, i.e., spoken form. It has a wide range of practical applications. Probably every one of you has already come across some of them. Ever used spelling correction? Or wondered about some strange hyphenation in a newspaper article? This is computational morphology at work. To solve such seemingly simple tasks often poses hard problems for a computer program. This section shall provide you with some insights into why this is so and what techniques are available to tackle these tasks.

1 Introduction

Natural languages have intricate systems to create words and word forms from smaller units in a systematic way. The part of linguistics dealing with these phenomena is morphology. This chapter starts with a quick overview over this fascinating field. It continues with applications of computational morphology. The rest is devoted to processing techniques. Computational morphology has evolved from very modest beginnings using full form lexica or some ad-hoc concatenation techniques to the much more powerful tools available today. The chapter concludes with a number of examples for encoding morphological phenomena from different languages using these tools.

2 Linguistic fundamentals

What is morphology all about? A simple answer is that morphology deals with words. In formal language words are just arbitrary strings denoting constants or variables. Nobody would care about a morphology of formal languages. In natural languages the picture is very different. Every human language contains some hundred thousands of words. And continuously new words are integrated while others are drifting out of use. This infinity of words is produced from a finite collection of smaller units. The task of morphology is to find and describe the mechanisms behind this process.

The basic building blocks in morphology are MORPHEMES. They are defined as the smallest unit in language to which a meaning may be assigned or, alternatively, as the minimal unit of grammatical analysis. Morphemes are abstract entities that express basic features. Either semantic concepts denoting entities or relationships in our world like *door*, *blue* or *take*. Such morphemes are called roots. Or syntactic features like *past* or *plural*.

Their realisation as part of a word is called MORPH. Often, there is a one-to-one relation, e.g., the morpheme *door* is realized as the morph *door*. With *take*, on the other hand, we find the two possibilities *take* and *took*. In such a case we speak of allomorphs. Plural in English is usually expressed by the morph *-s*. There are exceptions though: in *oxen* plural is expressed through the morph *-en*, in *men* by stem vowel alteration. All these different forms are allomorphs of the plural morpheme.

A basic distinction is the one between bound and free morphs. A free morph may form a word on its

own, e.g., the morph *door*. We call such words monomorphemic because they consist of a single morph. Bound morphs, on the other hand, occur only in combination with other forms. All affixes are bound morphs. For example, the word *doors* consists of the free morph *door* and the bound morph *-s*. Words may also consist of free morphs only, e.g., *tearoom*, or bound morphs only, e.g., *aggression*.

Every language typically contains some ten thousand morphs. This is a magnitude below the number of words. Strict rules govern the combination of these morphs to words (cf. 2.4). This way of structuring the lexicon makes the cognitive load of remembering so many words much easier.

2.1 What is a word?

Surprisingly, there is no easy answer to this question. One can easily spot „words" in a text because they are separated from each other by blanks or punctuation. However, if you record ordinary speech you will find out that there are no breaks between words. But, we could isolate units which occur over and over again in speech, but in different combinations. So the notion of „word" makes sense. But how do we define it?

We may look at „words" from different perspectives. To syntax „words" are the units that make up sentences. Words are grouped according to their function in the sentential structure. Each group gets a tag—usually called part-of-speech or word category—and grammar deals with these tags only, omitting the details of specific words.

Morphology, on the other hand, is concerned with the inner structure of „words". It tries to uncover the rules that govern the formation of words from smaller units. We notice that words that convey the same meaning look differently depending on their syntactic context. Take, e.g., the words *degrade*, *degrades*, *degrading*, and *degraded*. We can think of those as different forms of the same „word". We call the part that carries the meaning of those forms a base form. In our example this is the form *degrade*. All other forms in this example are produced by adding a suffix. A wide range of other possibilities will be shown in [section 2.3](#). All the different forms of a word together are called its *paradigm*. The part of morphology governing the production of these forms is called [inflection](#).

Base forms in English are at the same time always word forms in their own right, e.g., the base form *degrade* is also present tense, active voice, non 3rd person singular. In other languages we find a slightly different situation. In *Italian* nouns are marked for gender and number. Different affixes are used to signal masculine and feminine on the one hand and singular and plural on the other hand.

	SINGULAR	PLURAL	
MASCULINE	pomodor <u>o</u>	pomodor <u>i</u>	‘tomato’
FEMININE	cipoll <u>a</u>	cipoll <u>e</u>	‘onion’

Neither of the two forms of a noun can function as the base form. Instead, we must assume that the base form is what is left over after removing the respective suffixes, i.e., *pomodor-* and *cipoll-*. Such base forms that cannot occur as word forms in their pure form are called stems.

Base forms themselves are not necessarily atomic. By comparing *degrade* to *downgrade*, *retrograde* and *upgrade* on the one hand and *decompose*, *decrease* and *deport* on the other hand we can see that it is composed of the morphs *de-* and *grade*. The morpheme carrying the central meaning is often called the root of the word. A root may combine with suffixes (cf. 2.2.2.2) or other roots (cf. 2.2.2.3) to form new base forms.

Finally, we can describe „word" from a phonological perspective. Important for morphology is that phonological units define the range for phonological processes. Often, the phonological word is identical to the morphological word but sometimes boundaries differ. For example, the morphophonological process of final devoicing in German (cf. 2.3.2.2) works on syllable structure. Let's look at two words

derived from the root *lieb*. The word *be+lieb+ig* (arbitrary) is realized as /b'liːbik/ because it is a single phonological word. On the other hand, *lieb+lich* (lovely) is realized as /liːpliC/. Here, the last consonant of the root is devoiced because the two morphs are separated by a phonological word boundary.

2.2 Functions of morphology

How much and what sort of information is expressed by morphology differs widely between languages. Information that in some languages is expressed by syntax is expressed morphologically in others. Take, e.g., the expression of future tense: English uses an auxiliary verb construction, Spanish a suffix.

I speak - hablo
I will speak - hablaré

Also, some type of information may be present in one language while and missing in another. In many languages plural marking for nouns is mandatory. In Japanese it is absent.

book - hon
books - hon

The means for encoding information also vary widely. Most common is the use of different types of affixes. Traditionally, linguists discriminate between the following types of languages with regard to morphology:

- Isolating languages (e.g. Mandarin Chinese): there are no bound forms, e.g., no affixes that can be attached to a word. The only morphological operation is composition.
- Agglutinative languages (e.g. Ugro-Finnic and Turkic languages): all bound forms are either prefixes or suffixes, i.e., they are added to a stem like beads on a string. Every affix represents a distinct morphological feature. Every feature is expressed by exactly one affix.
- Inflectional languages (e.g. Indo-European languages): distinct features are merged into a single bound form (a so-called portmanteau morph). The same underlying feature may be expressed differently, depending on the paradigm
- Polysynthetic languages (e.g. Inuit languages): these languages express more of syntax in morphology than other languages, e.g., verb arguments are incorporated into the verb.

This classification is quite artificial. Real languages rarely fall cleanly into one of the above classes, e.g., even Mandarin has a few suffixes. Moreover, this classification mixes the aspect of what is expressed morphologically and the means for expressing it.

2.2.1 Inflection

Inflection is required in particular syntactic contexts. It does not change the part-of-speech category but the grammatical function. The different forms of a word produced by inflection form its paradigm. Inflection is *complete*, i.e., with rare exceptions all the forms of its paradigm exist for a specific word. Regarding inflection, words can be categorized in three classes:

- Particles or not-inflecting words: they occur in just one form. In English, prepositions, adverbs, conjunctions and articles are particles;
- Verbs or words following conjugation;
- Nominals or words following declination, i.e., nouns, adjectives, and pronouns.

Conjugation is mainly concerned with defining tense and aspect and agreement features like person and number. Take for example the *German* verb 'lesen' (to read). German verb forms come in present and past tense, indicative or subjunctive.

	PRESENT				PAST			
	INDICATIVE		INDICATIVE		SUBJUNCTIVE		SUBJUNCTIVE	
	SINGULAR	PLURAL	SINGULAR	PLURAL	SINGULAR	PLURAL	SINGULAR	PLURAL
1 st PERSON	lese	lesen	lese	lesen	las	lasen	läse	läsen
2 nd PERSON	liest	lest	lesest	leset	last	last	läsest	läset
3 rd PERSON	liest	lesen	lese	lesen	las	lasen	läse	läsen
PARTICIPLE	lesend				gelesen			
IMPERATIVE	lies	lest						
INFINITIVE	lesen							

Declination marks various agreement features like number (singular, plural, dual, etc.), case (as governed by verbs and prepositions, or to mark various kinds of semantic relations), gender (male, female, neuter), and comparison.

2.2.2 Derivation and Compounding

In contrast to inflection which produces different forms of the same word derivation and compounding are processes that create *new words*. Thus, derivation and compounding have nothing to do with morphosyntax. They are a means to extend our lexicon in an economic and principled way.

In derivation, a different word--often of a different part-of-speech category--is produced by adding a bound morph to a stem. Derivation is incomplete, i.e., a derivational morph cannot be applied to all words of the appropriate class. For example, in German the very productive derivational suffix *-bar* can be applied to many but not all verbs to produce adjectives:

essen	‘to drink’	- <u>essbar</u>	‘eatable’
hören	‘to hear’	- <u>hörbar</u>	‘audible’
absehen	‘to conceive’	- <u>absehbar</u>	‘conceivable’
sehen	‘to see’	- * <u>sehbar</u>	‘visible’

Application of a derivational morpheme may be restricted to a certain subclass. For example, application of the English derivational suffix *-ity* is restricted to stems of Latin origin, while the suffix *-ness* can apply to a wider range:

rare	- <u>rarity</u>	- ? <u>rareness</u>
red	- * <u>reddity</u>	- <u>redness</u>
grave	- <u>gravity</u>	- <u>graveness</u>
weird	- * <u>weirdity</u>	- <u>weirdness</u>

Derivation can be applied recursively, i.e., words that are already the product of derivation can undergo the process again. That way a potentially infinite number of words can be produced. Take, for example, the following chain of derivations:

hospital — hospitalize — hospitalization — pseudohospitalization

Semantic interpretation of the derived word is often difficult. While a derivational suffix can usually be given a unique semantic meaning many of the derived words may still resist compositional interpretation. This may be due to lexicalization, i.e. a form is no more transparent because, or ambiguity of the underlying base form. For a more detailed discussion see Trost (1993).

While inflectional and derivational morphology are mediated by the attachment of a bound morph to a base form, compounding is the joining of two or more base forms to form a new word. Most common is just setting two words one after the other, as in *state monopoly*, *bedtime* or *red wine*. In some cases parts are joined by a linking morphem (usually the remnant of case marking) as in *bull's eye* or German *Liebeslied* (love-song).

The last part of a compound usually defines its morphosyntactic properties. Semantic interpretation is even more difficult than with derivation. Almost any semantic relationship may hold between the components of a compound:

Wienerschnitzel 'cutlet made in Viennese style'
Schweineschnitzel 'cutlet made of pork'
Kinderschnitzel 'cutlet made for children'

The boundary between derivation and compounding is fuzzy. Historically, most derivational suffixes developed from words frequently used in compounding. An obvious example is the *-ful* suffix as in *hopeful*, *wishful*, *thankful*.

Phrases and compounds cannot always be distinguished. The English expression *red wine* in its written form could be both. In spoken language the stress pattern differs: *red wíne* vs. *réd wine*. In German phrases are morphologically marked, while compounds are not: *roter Wein* vs. *Rotwein*. But for verb compounds the situation is similar to English: *zu Hause bleiben* vs. *zuhausebleiben*.

2.3 What constitutes a morph?

Every word form must at the core contain some root form. This root can (must) then be complemented with additional morphs. How are morphs realized? Obviously, a morph must somehow be recognizable in the phonetic or orthographic pattern constituting the word. The most common type of morph is a continuous sequence of phonemes. All roots and affixes are of this form. A complex word can then be analyzed as a series of morphs concatenated together. Agglutinative languages function almost exclusively this way. But there are surprisingly many other possibilities.

2.3.1 Affixation

An affix is a bound morph that is realised as a sequence of phonemes (or graphemes). The by far most common types of affixes are prefixes and suffixes. Many languages have only these two types of affixes. Among them is English (at least under standard morphological analyses).

A prefix is an affix that is attached in front of a stem. An example is the English negative marker *un-* attached to adjectives:

common uncommon

A suffix is an affix that is attached after a stem. Take, e.g., the English plural marker *-s*:

shoe shoes

Across languages suffixation is far more frequent than prefixation. Also, certain kinds of morphological information are never expressed via prefixes, e.g., nominal case marking. Many computational systems for morphological analysis and generation assume a model of morphology based on prefixation and suffixation only.

A circumfix is the combination of a prefix and a suffix which together express some feature. Both

theoretically and from a computational point of view a circumfix can be viewed as really two affixes applied one after the other.

In *German*, the circumfixes *ge--t* and *ge--n* form the past participle of verbs:

sagen ‘to say’ gesagt ‘said’
laufen ‘to run’ gelaufen ‘run’

An infix is an affix where the placement is defined in terms of some phonological condition(s). These might result in the infix appearing within the root to which it is affixed. In *Bontoc*, a Philippine language, the infix *-um-* turns adjectives and nouns into verbs (Fromkin and Rodman 1983). The infix attaches after the initial consonant:

/fikas/	‘strong’	/f <u>um</u> ikas/	‘to be strong’
/lalad/	‘red’	/l <u>um</u> ilad/	‘to be red’
/fuzul/	‘enemy’	/f <u>um</u> uzul/	‘to be an enemy’

Reduplication is a border case of affixation. The form of the affix is a function of the stem to which it is attached, i.e., it copies (some portion of) the stem. Reduplication may be complete or partial. In the latter case it may be prefixal, infixal or suffixal. Reduplication can include phonological alteration on the copy or the original.

In *Javanese* complete reduplication is used to express the habitual-repetitive. In case the second vowel is non-/a/, the first vowel in the copy is made nonlow (changing /a/ to /o/ and /E/ to /e/) and the second becomes /a/. When the second vowel is /a/, the copy remains unchanged while in the original the /a/ is changed to /E/ (Kiparsky 1987):

/adus/	‘take a bath’	/odasadus/
/bali/	‘return’	/obalabali/
/bozən/	‘tired of’	/bozanbozən/
/ələq/	‘return’	/eləqələq/
/dolam/	‘recreate’	/dolandoləm/
/udan/	‘horse’	/udanudan/

Partial reduplication is more common. In *Yidin’*, an Australian language, prefixal reduplication is used for plural marking. Reduplication involves copying the ‘minimal word’ (Nash 1980).

/mulari/	‘initiated man’	/mulamulari/
/gindalba/	‘lizard’	/gindalgindalba/

An example for infixal reduplication is the frequentative in *Amharic*, a semitic language spoken in Ethiopia (Rose 2000).

/kətəfə/	‘chop’	/kətətəfə/	‘chop a lot’
/k’əbələ/	‘decrease’	/k’ibabələ/	‘decrease greatly’
/wək’ət’ə/	‘fight’	/wik’ak’ət’ə/	‘fight a lot’
/lək’ət’ə/	‘mix’	/lik’ak’ət’ə/	‘mix a lot’

From a computational point of view one property of reduplication is especially important: Since reduplication involves copying it cannot—at least in the general case—completely be described with the use of finite-state methods.

2.3.2 Root-and-template morphology

Semitic languages (at least according to standard analyses) exhibit a very peculiar type of morphology: A so-called root, consisting of two to four consonants, conveys the basic semantic meaning. A vowel

pattern marks information about voice and aspect. A derivational template gives the class of the word (traditionally called *binyan*).

In *Arabic* verb stems are constructed this way. The root *ktb* (write) produces--among others--the following stems:

Template	Vowel pattern		
	a (active)	ui (passive)	
CVCVC	katab	kutib	'write'
CVCCVC	kattab	kuttib	'cause to write'
CVVCVC	ka:tab	ku:tib	'correspond'
tVCVVCVC	taka:tab	tuku:tib	'write each other'
nCVVCVC	nka:tab	nku:tib	'subscribe'
CtVCVC	ktatab	ktutib	'write'
stVCCVC	staktab	stuktib	'dictate'

2.3.3 Modification in phonetic substance

This term subsumes processes which do neither introduce new nor remove existing segments. Morphs are not realized as any string of phonemes, but as a change of phonetic properties or an alteration of the prosodic shape.

Ablaut refers to vowel alternations inherited from Indo-European. It is a pure example of vowel modification as a morphological process. Examples are strong verbs in Germanic languages like English (e.g., *swim* — *swam* — *swum*). In *Icelandic* this process is still more common and more regular than in most other Germanic languages. The following example is from Sproat (1992, p.62):

STEM	PAST SING.	PAST PL.	PPP	
/bit/	/beit/	/bit/	/bit/	'to bite'
/ri:f/	/reif/	/rif/	/rif/	'to tear'

Umlaut has its origin in a phonological process, whereby root vowels were assimilated to a high-front suffix vowel. When this suffix vowel was lost later on, the change in the root vowel became the sole remaining mark of the morphological feature originally signalled by the suffix.

In *German* the plural of nouns may be marked by umlaut (sometimes in combination with a suffix), whereby in the stem vowel the feature *back* is changed to *front*:

SINGULAR		PLURAL		
Mutter	/mʊtɐ/	Mütter	/mʏtɐ/	'mother'
Garten	/gɑ:tn̩/	Gärten	/gɛ:tn̩/	'garden'
Hof	/hɔ:f/	Höfe	/hø:fə/	'yard'

Another possibility to realize a morpheme is to alter the prosodic shape. Tone modification can be used to signal certain morphological features.

In *Ngbaka*, spoken in the Democratic Republic of Congo, tense-aspect contrasts are expressed by four different tonal variants (Nida 1949):

LOW	MID	LOW-HIGH	HIGH	
há	h̄á	h̄á	há	'put more than one thing'
h̄kpóló	h̄kpóló	h̄kpóló	h̄kpóló	'return'
h̄'ílí	h̄'ílí	h̄'ílí	h̄'ílí	'cut'

A morpheme may be realised by a stress shift. *English* noun-verb derivation sometimes uses a pattern where the stress is shifted from the first to the second syllable:

NOUN	VERB
éxport	expórt
récord	recórd
cónvict	convíct

2.3.4 Suppletion

Total modification is a process occurring sporadically and idiosyncratically within inflectional paradigms. It is usually associated with forms that are used very frequently. Examples in English are *went*, the past tense of *go*, and the forms of *to be*: *am*, *are*, *is*, *was* and *were*.

2.3.5 Zero Morphology

Sometimes a morphological operation has no phonological expression whatsoever. Examples are found in many languages.

English noun-to-verb derivation is often not explicitly marked:

man The man smiled. Man the boats.

house He buys a house. They house in a cave.

A possible analysis is to assume a zero morph which attaches to the noun to form a verb: book+ \emptyset_V . Another possibility is to assume two independent lexical items disregarding any morphological relationship.

2.4 The structure of words: Morphotactics

Somehow morphs must be put together to form words. A word grammar is determining the way this has to be done. This part of morphology is called *morphotactics*. As we have seen, the most usual way is simple concatenation. Let's have a look at the constraints involved. What are the conditions governing the ordering of morphemes in *pseudohospitalization*?

(1) *hospitalationizepseudo, *pseudoizehospitalation

(2) *pseudohospitalationize

In (1) an obvious restriction is violated: *pseudo-* is a prefix and must appear ahead of the stem, *-ize* and *-ation* are suffixes and must appear after the stem. The violation in (2) is less obvious. In addition to the pure ordering requirements there are also rules governing to which types of stems an affix may attach: *-ize* attaches to nouns and produces verbs, *-ation* attaches to verbs and produces nouns.

One possibility to describe the word formation process is to assume a functor-argument structure. Affixes are functors that pose restrictions on their (single) argument. That way a binary tree is constructed. Prefixes induce right branching and suffixes left branching.

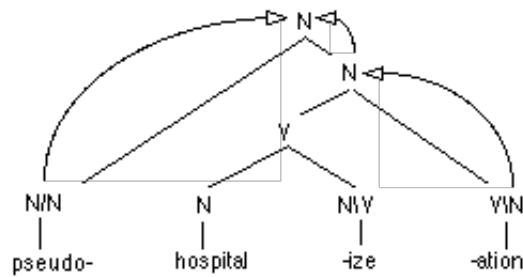


Fig. 1: The internal structure of the word *pseudohospitalization*

In figure 1 the functor *pseudo-* takes a nominal argument to form a noun, *-ize* a nominal argument to form a verb, and *-ation* a verbal argument to form a noun. This description renders two different possible structures for *pseudohospitalization*. The one given in figure 1 and a second one where *pseudo-* combines first directly with *hospital*. We may or may not accept this ambiguity. To avoid the second reading we could state a lexical constraint that a word with the head *pseudo-* cannot serve as an argument anymore.

2.4.1 Constraints on affixes

Affixes is that they attach to specific categories only. This is an example for a syntactic restriction. Restrictions may also be of a phonological, semantic or purely lexical nature. A semantic restriction on the English adjectival prefix *un-* prevents its attachment to an adjective that already has a negative meaning:

unhappy *unsad
 unhealthy *unill
 unclean *undirty

The fact that in English some suffixes may only attach to words of Latin origin (cf. 2.2.2) is an example for a lexical restriction.

2.4.2 Morphological vs. phonological structure

In some cases there is a mismatch between the phonological and the morphological structure of a word. One example is comparative formation with the suffix *-er* in English. Roughly, there is a phonological rule that prevents attaching this suffix to words that consist of more than two syllables:

great greater
 tall taller
 happy happier
 competent *competenter
 elegant *elegantier

If we want to stick to the above rule *unrulier* has to be explained with a structure where the prefix *un-* is attached to *rulier*. But, from a morphological point of view, the adjective *ruly* does not exist, only the negative form *unruly*. This implies that the suffix *-er* is attached to *unruly*. We end up with an obvious mismatch!

Another potential problem is cliticization. A clitic is a syntactically separate word phonologically realized as an affix. The phenomenon is quite common across languages.

- In English auxiliaries have contracted forms that function as affixes:

he shall return -> *he'll return*

- In German prepositions can combine with the definite article
an dem Tisch -> *am Tisch*
in das Haus -> *ins Haus*
- In Italian personal pronouns can be attached to the verb. In this process the ordering of constituents is also altered.
ce ne facciamo -> *facciamocene*

2.5 The Influence of Phonology

Morphotactics is responsible to govern the rules for the combination of morphs into larger entities. One could assume that this is all a system needs to know to break down words into their component morphemes. But there is another aspect that makes things more complicated: Phonological rules may apply and change the shape of morphs. To deal with these changes and their underlying reasons is the area of morphophonology.

2.5.1 Phonology vs. orthography

Most applications of computational morphology deal with text rather than speech. But, written language is rarely a true phonemic description. For some languages, e.g., Finnish, Spanish or Turkish orthography is a good approximation for a phonetic transcription. English, on the other hand, has very poor correspondence between writing and pronunciation. As a result, we often have to deal with orthography rather than phonology. A good example are English plural rules (cf. 2.4.1).

2.5.2 Local phenomena

We have shown that, by and large, words are composed by concatenating morphs. In many cases this concatenation process will induce some phonological change in the vicinity of the morph boundary.

Assimilation is a process where the two segments at a morph boundary influence each other, resulting in some feature change that makes them more similar. Take, for example, the English *in-* prefix where the *n* changes to *m* before labials:

<in+feasible> -> infeasible
<in+mature> -> immature
<in+probable> -> improbable
<in+secure> -> insecure

Another possibility is epenthesis (insertion) or elision (deletion) of a segment under certain (phonological) conditions. Take for example the English plural formation:

<cat+s> -> cats
<door+s> -> doors
<dish+s> -> dishes
<bliss+s> -> blisses
<match+s> -> matches
<fox+s> -> foxes

In this case the rule requires the insertion of an /' / between /s/, /z/, /S/, or /Z/ and another /s/. On the other hand, in German the suffix —st attached to stems ending in /s/ loses its starting segment /s/:

<leb+st> -> lebst
 <sag+st> -> sagst
 <ras+st> -> rast
 <trotz+st> -> trotzt
 <hex+st> -> hext

We see that the change is not purely phonologically motivated. The same condition, namely two adjoining /s/ phonemes leads to different results: Either the epenthesis of an /'/ between the two, or the elision of the second /s/. Moreover, the notion of insertion or deletion is purely descriptive. Phonological theory may explain the underlying processes completely different. Nonetheless, this is the view most often taken by work in computational morphology.

2.5.3 Long-distance effects

Most common is vowel harmony but there are also examples of consonant harmony. Vowel harmony is a phonological process where the leftmost (in rare cases the rightmost) vowel in a word influences all the following (preceding) vowels. Among the languages exhibiting vowel harmony are Finnish, Hungarian, Turkic and many African languages.

Let's have a look at vowel harmony in *Turkish*. The nine vowels of the Turkish language can be specified the following way:

	i	ɪ	ü	u	e	a	ö	o
HIGH	+	+	+	+				
BACK		+			+	+		+
ROUND			+	+			+	+

Turkish has two different harmony rules, called *small* and *large* respectively:

e	i	ö	ü	->	e	e	i	->	i
						ö	ü	->	ü
a	ɪ	o	u	->	a	a	ɪ	->	ɪ
						o	u	->	u

Only the stem vowel in a word is lexically determined. All the following vowels are realized in accordance to the harmony rules. For example, the root *ev* (house) induces either an *e* or an *i* in the attached suffixes. Which one of the two is realized is a property of the respective suffix.

NOM.	yil	kız	gül	pul	ev	sap	köy	son
GEN.	yilin	kızın	gülün	pulun	evin	sapın	köyün	sonun
PLURAL	yillar	kızlar	güller	pullar	evler	saplar	köyler	sonlar
GEN. PL.	yillerin	kızların	güllerin	pulların	evlerin	sapların	köylerin	sonların
gloss	'year'	'girl'	'rose'	'stamp'	'house'	'stalk'	'village'	'end'

In this example the plural suffix follows the „small" harmony and the genitive suffix the „large" harmony rule.

3 Applications of Computational Morphology

Computational morphology has many practical applications. Besides low-level applications, computational morphology contributes to many speech and language processing systems.

3.1 Low-level applications

Hyphenation is almost exclusively done automatically. Although the task seems at first glance extremely simple only a human expert can achieve a 100% success rate. Segmenting words correctly into their morphs helps to solve the task. The major problem are spurious segmentations.

Spelling correction is another low-level application. Just comparing input against a list of word forms has a number of drawbacks. Such a list will never contain all the words occurring in a text and enlarging the list has the negative side effect of including more and more obscure words that will match with typos thus preventing their detection. Most systems use a root lexicon, plus a relatively small set of affixes and simple rules to cover morphotactics.

Stemmers are used in information retrieval to reduce as many related words and word forms as possible to a common canonical form which can then be used in the retrieval process. One should note that this canonical form is not necessarily the base form. The main requirement is—like in all the above tasks—robustness.

Another application is to segment text in Chinese, Japanese or Korean. In these languages words in a sentence are not separated by blanks or punctuation marks. Morphological analysis can be used to perform the task of word separation.

A related problem is the inputting of Japanese text. Japanese is written with a combination of two independent character sets. Kanji, the morphemic Chinese characters are used for open-class morphemes (verbs, nouns and adjectives). Kana has (about 50) syllabic characters and is mainly used for closed-class morphemes although in principle all Japanese words can be written exclusively in kana.

Since there are several thousand kanji characters, many Japanese text input systems use kana-kanji conversion. The text is typed in kana and the relevant portions are subsequently converted to kanji. The mapping from kana to kanji is quite ambiguous. A combination of statistical and morphological methods is applied to solve that task.

3.2 Natural language applications

An obvious application area for morphological components are more general natural language processing systems involving parsing and/or generating natural language utterances in written or spoken form. There is a wide range of such applications from message and information extraction to dialog systems and machine translation. For many current applications, only inflectional morphology is considered.

In a parser, morphological analysis of words is an important prerequisite for syntactic analysis. Properties of a word the parser needs to know are its part-of-speech category and the morphosyntactic information encoded in the particular word form. Another important task is lemmatization, i.e., finding the corresponding dictionary form for a given input word, because for many applications a lemma lexicon is used to provide more detailed syntactic (e.g, valency) and semantic information for a deep analysis.

In generation, on the other hand, the task is to produce the correct word form from the base form plus the relevant set of morphosyntactic features.

3.3 Speech applications

A text-to-speech system takes (electronically stored) text as input and produces speech from it. Morphological analysis helps to solve two different tasks in such systems. One is to guide the grapheme-to-phoneme conversion. Characters are often ambiguous with respect to their translation into phonemes. Finding out the underlying morphological structure is necessary for solving the task correctly. The sequence *th*, is usually pronounced as /D/ or /T/ in English. In the word *hothouse* we need to know the morph structure <hot+house> to correctly pronounce the *th* sequence as /th/.

A less obvious application is the use of morphological analysis to help in determining the part-of-speech category of words. This is an important prerequisite of syntactic analysis which is the basis for coming up with a correct prosody.

Speech recognition is a field where morphological analysis will become ever more important. At the moment most available systems make use of full form lexicons and perform their analysis on a word basis. Increasing demands on the lexicon size on the one hand and the need to limit the necessary training time on the other hand will make morph-based recognition systems more attractive.

4 Computational Morphology

The most basic task in computational morphology is to take a string of characters or phonemes as input and deliver an analysis as output. The input could, for example be the English word form in (1). One possible output could be the string of underlying morphemes as in (2), another one a morphosyntactic interpretation as in (3).

1. incompatibilities
2. in+con+patible+ity+s
3. incompatibility+NounPlural

Let's start with the task of mapping (1) to (3). The easiest way to achieve a result is to have a long list of pairs where the left side represents some word form and the right side its interpretation. This is basically the notion of full form lexicon. Its advantages are simplicity and applicability to all possible phenomena. The main disadvantages are redundancy and inability to cope with forms not contained in the lexicon.

Less redundant are so-called lemma lexica. A lemma is a canonical form taken as the representative for all the different forms of a paradigm. Usually, the base form is selected as this canonical form. An interpretation algorithm relates every form to its lemma plus delivering a morphosyntactic interpretation. As a default, forms are expected to be string concatenations of base form (= lemma) and affixes. Affixes must be stored in a separate repository together with the relevant morphotactic information about how they may combine with other forms. Interpretation then simply means finding a sequence of affixes and a base form that conforms to morphotactics. For different reasons a given word form may not conform to this simple picture:

- With very frequently used words we often find suppletion, e.g., *to go* has the completely unrelated form *went*.

One clearly needs some exception handling mechanism to cope with suppletion. A possible solution is to have secondary entries where you store suppleted forms together with their morphosyntactic information. These secondary forms are then linked to the corresponding primary form, i.e., the lemma.

- Morphs are realised in a non-concatenative way, e.g., tense of strong verbs in English: *give* — *gave* - *given*, *find* - *found* — *found*

In languages like English, where these phenomena affect only a fairly small and closed set of words these forms can be treated like suppletion. Alternatively, some exception handling mechanism (usually developed ad-hoc and language-specific) is applied.

- Due to phonological rules a word form may exhibit some change in shape, e.g., in English suffixes starting with *s* (plural of nouns, 3rd person marker, superlative marker) may not directly follow stems ending in a syllabiant (e.g., *dish* — *dishes*)

If morphophonological processes in a language are few and local the lemma lexicon approach can still be successful. In our example it suffices to assume two plural endings: *-s* and *-es*. For all base forms it must be specified whether the former or the latter of the two endings may be attached.

Apart from the obvious limitations with regard to the treatment of morphophonological rules on a more general scale the approach has some other inherent restrictions.

- The algorithm is geared towards analysis. For generation purposes, one needs a completely different algorithm and data.
- Interpretation algorithms are language-specific because they encode both the basic concatenation algorithm and the specific exception-handling mechanism.
- The approach was developed for morphosyntactic analysis. An extension to handle more generally the segmenting of word forms into morphs is difficult to achieve.

4.1 Finite-state Morphology

Because most morphological phenomena can be described with regular expressions the use of finite-state techniques for morphological components is common. In particular, when morphotactics is seen as a simple concatenation of morphs it can straightforwardly be described by a finite automata.

It was not so obvious though how to describe non-concatenative phenomena like vowel harmony, root-and-template morphology or infixation in such a framework.

4.1.1 Two-level morphology

In this section we describe a system where morphophonology is taken care of by a separate mechanism that is well integrated with the morphotactical component. It has the further advantages of being non-directional (applicable to analysis and generation) and language-independent (because of its purely declarative specification of language-specific data).

Rules for the description of morphophonological phenomena are standard in generative phonology. There, the derivation of a word form from its lexical structure is performed by the successive application of phonological rules creating a multi-step process involving several intermediate levels of representation. Such an approach may be suited for generation but leads to problems if applied to analysis. Since the ordering of rule application influences the result it is difficult to reverse the process.

Several proposals were made on how to restrict rules and their application to overcome these problems. Two-level morphology is an attempt to overcome these problems. Originally proposed by Kimmo Koskenniemi (1984) it has since been implemented in a number of different systems and applied to a wide range of natural languages.

4.1.1.1 Two-level rules

As the name suggests two levels--called lexical level and surface level--suffice to describe the phonology (or orthography) of a natural language. On the surface level words appear just as they are pronounced (or written) in ordinary language, with the important exception of the null character which will be described later on. On the lexical level, the alphabet includes special symbols--so-called diacritics--which are mainly used to represent features that are no phonemes (or graphemes) but nevertheless constitute necessary phonological information. The diacritics '+' and '#' are used to indicate morph and word boundary respectively.

The two levels are linked by a set of pairs of lexical and surface characters constituting possible mappings between lexical and surface characters. Pairs are written as lexical character - colon - surface character (e.g. *a:a* or *+:0*). To any of these pairs rules may be attached to restrict their applicability.

Pairs with no attached rules are applied by default. Rules serve to licence the application of a pair in a certain phonological context. They are viewed as constraints on the mapping between the surface and the lexical form of morphs. Accordingly, they are applied in parallel and not one after the other like in generative phonology. Since no ordering of the rules is involved this is a completely declarative way of description.

A rule consists of the following parts:

- A substitution that indicates the affected character pair.
- left and right context define the phonological conditions for the substitution.
- One of four available operators defines the status of the rule: The context restriction operator \leq makes the substitution of the lexical character obligatory in the context defined by that rule (other phonological contexts are not affected). The surface coercion operator \Rightarrow restricts the substitution of the lexical character to exactly this context (it may not occur anywhere else). The \Leftrightarrow is a combination of the former two, i.e., the substitution must take place in exactly this context and nowhere else. The fourth operator $/\leq$ states prohibitions, i.e., the substitution may not take place in this context.

Let's look at a simple epenthesis rule:

(1a) $+ : e \leq s \ x \ z \ [\{ s \ c \} \ h] : _ s ;$

It specifies that a lexical morph boundary (indicated by '+') between *s*, *x*, *z*, *sh*, or *ch* on the left side and an *s* on the right side must correspond to surface level *e*. By convention a pair with identical lexical and surface character may be denoted by just a single character. Curly brackets indicate a set of alternatives, square brackets a sequence.

Rule (1a) makes no statements about other contexts where '+' may map to an 'e'. The rule covers some of the cases where an 'e' is inserted between stem and an inflectional morph starting with 's' (plural morpheme, 3rd person marker, superlative) in English. By default a morph boundary will map to the null character, but in the given specific context it maps to 'e'. The following example shall demonstrate the application of this rule (Vertical bars denote a default pairing, numbers the application of the corresponding rule):

```
#bliss+s# #fox+s# #dish+s# #watch+s#
| | | | | 1 | | | | | 1 | | | | | 1 | | | | | 1 | |
0blisses0 0foxes0 0dishes0 0watches0
```

Obviously, (1a) does not capture all the cases where epenthesis of 'e' occurs. For example, the forms *spies*, *shelves* or *potatoes* are not covered. A more complete rule is:

(1b) $+ : e \Leftrightarrow \{ s \ x \ z \ [\{ s \ c \} \ h : h] : v \ [C \ y :] \ [C \ o] \} _ s ;$

Formally, rule (1b) defines exactly all the contexts where '+' maps to an 'e' (because of the use of the \Leftrightarrow operator). It also makes use of some additional writing conventions. A colon followed by a character denotes the set of all pairs with that surface character. Accordingly, a character followed by a colon means the set of all pairs with that lexical character. Sets of characters can be globally defined and given names. The C stands for the set of English consonants (i.e., b:b, c:c, d:d,...). To cope with the *spies* example we need another rule which licences the mapping from 'y' to 'i'.

(2) $y : i \Leftrightarrow C _ \{ + : e \ [+ : e] \} ;$
 $\vee C + _ + : C ;$

Rule (2) specifies two distinct contexts. If either of them is satisfied the substitution must occur, i.e.,

contexts are OR-connected. The '+' operator in the second context indicates *at least one occurrence* of the preceding sign (accordingly, the operator '*' has the reading *arbitrarily many occurrences*). V stands for the set of vowels. Rules (1) and (2) in combination now correctly map *spies* with *spy+s*. Jointly with rule (3) for the mapping from 'f' to 'v' (1) takes also care of forms like *shelves* and *potatoes*:

$$(3) f:v \leq \{ e l \} _ +: s ; \\ \quad V _ e +: s ;$$

Let's see how the three rules interact to produce the expected results:

```
#spy+s# #toy+s# #shelf+s# #wife+s# #potato+s#
|||21|| | ||||| | |||||31|| | |||3||| | |||||1||
0spies0 0toy0s0 0shelves0 0wive0s0 0potatoes0
```

A given pair of lexical and surface strings can only map if they are of equal length. There is no possibility of omitting or inserting a character in one of the levels. On the other hand, elision and epenthesis are common phonological phenomena. To cope with these, the null character (written as 0) is included in both the surface and the lexical alphabet. The null character is taken to be contained in the surface string for the purpose of mapping lexical to surface string and vice versa but it does not show up in the output or input of the system. Diacritics are mapped to the null character by default. Any other mapping of a diacritic has to be licensed by a rule.

Assumption of the explicit null character is essential for processing. A mapping between a lexical and a surface string presupposes that for every position a character pair exists. This implies that both strings are of equal length (nulls are considered as characters in this respect). Rules can either be directly interpreted or compiled into finite state transducers. The use of finite state machinery allows for very efficient implementation. For a more in-depth discussion of implementational aspects consult chapter 37 and Beesley and Karttunen (2000).

One subtle difference between direct rule interpretation and transducers occurs in the repeated application of the same rule to one string. The transducer implicitly extends the phonological context to the whole string. It must therefore explicitly take care of overlapping right and left contexts (e.g., in (1) the pair *s:s* constitutes both a left and right context). With direct interpretation a new instance of the rule is activated every time the left context is found in the string and overlapping must not be treated explicitly.

4.1.1.2 The continuation lexicon

Up to now we have only described the rule part of two-level morphology which is responsible for taking care of morphonological phenomena. It is complemented by a partitioned lexicon of morphs (or words) that takes care of word formation by affixation. The lexicon consists of (non-disjunctive) sublexica, so-called continuation classes. For every morph, a set of legal continuation classes is specified. This set defines which sublexicon must be searched for continuations. The class of morphs which can start a word is stored in the so-called "*init lexicon*".

The whole process is equivalent to stepping through a finite automaton. A successful match can be taken as a move from some state *x* of the automaton to some other state *y*. Lexical entries can be thought of as arcs of the automaton: a sublexicon is a collection of arcs having a common *from* state.

The lexicon in two-level morphology is used for two purposes: one is to describe which combinations of morphs are legal words of the language, the other one is to act as a filter whenever a surface word form shall be mapped to a lexical form. Its use for the second task is crucial because otherwise there would be no way to limit the insertion of the null character.

To enable fast access, lexicons are organized in the form of a letter trie (Fredkin, 1960). Such a structure is well suited for an incremental (letter-by-letter) search because at every point in the trie exactly those continuations leading to legal morphs are available. With every node which represents a legal morph its continuation classes are stored. In recognition we can now make use of that structure. Search starts at the root of the trie. Each character which is proposed must be matched against the lexicon. Only if that character is a legal continuation at that node in the trie it may be considered as a possible mapping.

In recent implementations the lexicon and the two-level rules are collapsed into a single, large transducer, resulting in a very compact and efficient system

4.1.2 Related Formalisms

Black et al. (1987) note the inelegance of Koskeniemi's formalism when describing a phonological (or orthographic) change affecting sequences of characters. They propose a rule format consisting of a surface string (called LHS for *left hand side*), an operator (\langle or \triangleright) and a lexical string (called RHS for *right hand side*). LHS and RHS must be of equal length. Surface-to-lexical rules (\triangleright) request that there exists a partition of the surface string where each part is the LHS of a rule and the lexical string the concatenation of the corresponding RHSs. Lexical-to-surface rules (\langle) request that any substring of a lexical string which equals a RHS of a rule must correspond to the surface string of the LHS of the same rule. The rules in (4) are equivalent to rule (1a).

$$(4) \text{ ses} \Rightarrow \text{s+s} \quad \text{ses} \Leftarrow \text{s+s} \quad \text{shes} \Rightarrow \text{sh+s} \quad \text{shes} \Leftarrow \text{sh+s} \quad \text{xes} \Rightarrow \text{x+s} \quad \text{xes} \Leftarrow \text{x+s} \\ \text{zes} \Rightarrow \text{z+s} \quad \text{zes} \Leftarrow \text{z+s} \quad \text{ches} \Rightarrow \text{ch+s} \quad \text{ches} \Leftarrow \text{ch+s}$$

These rules collapse context and substitution into one undistinguishable unit. Instead of regular expressions only strings are allowed. One drawback is that surface-to-lexical rules may not overlap. If two different changes happen to occur close to each other they must be captured in a single rule. Also, long-distance phenomena like vowel harmony cannot be described in this scheme. Ruessink (1989) removes this problem by introducing contexts again. Both LHS and RHS may come with a left and right context. LHS and RHS may also be of different length, doing away with the null character. Though he gives no account of the complexity of his algorithm one can suspect that it is in general less constrained than the Koskeniemi system.

An inherently difficult problem for two-level morphology is the root-and-template morphology of Semitic languages. One solution is the introduction of multi-tape formalisms as first described in the seminal paper by Kay (1987). The best-documented current system is SEMHE described in Kiraz (1996, 1997). SEMHE is based on Ruessink's formalism with the extension of using three input tapes: one each for the root, the vowel pattern and the template.

Another extension to the formalism is realized in X2MorF (Trost 1992). In the standard system, morphologically motivated phenomena like umlaut must be described by introducing some pseudosegmental material in the lexical level (see, e.g., 2.4.3.3). In X2MorF an additional morphological context is available to describe such phenomena more naturally.

4.2 Alternative formalisms

Alternative proposals for morphological systems have been made in computational linguistics. They include so-called paradigmatic morphology described in Calder (1989) and the DATR system (Evans and Gazdar 1996). Common to both is the idea to introduce some

default mechanism which makes it possible to define a hierarchically structured lexicon where general information is stored at a very high level. Lower in the hierarchy this information can be overwritten. Both systems seem to be more concerned with morphosyntax than with morphonology. It is an open question if these approaches could somehow be combined with two-level rules.

4.3 Examples

4.3.1 Vowel harmony in Finnish

Finnish has eight vowels. They are classified into *back+* (a, o, u), *back-* (ä, ö, y) and neutral (e, i). In a Finnish word vowels must be either all back+ or all back- (disregarding neutral vowels).

$$V = \{a, o, u, \text{ä}, \text{ö}, y, e, i\}$$

$$V_b = \{a, o, u\} \quad V_f = \{\text{ä}, \text{ö}, y\}$$

$$[1] \quad \{A:a|O:o|U:u\} \quad P =:V_b =: (-V_f)^* _;$$

$$[2] \quad \{A:\text{ä}|O:\text{ö}|U:y\} \quad P \{ \# | =:V_f \} =: (-V_b)^* _;$$

```
#taivas+tA# #puhelin+tA# #syy+tA#
|||||1| |||||1| |||||2|
0taivas0ta0 0puhelin0ta0 0syy0tä0
```

4.3.2 Final devoicing in (spoken) German

Final devoicing is a morphophonological process where a voiced consonant is devoiced when it occurs in final position in the syllable. Take for example the root /rã:d/ (wheel). The singular form is realized as /ra:t/, while in the plural form /rẽ:dã/ the consonant stays voiced. This phenomenon is not reflected in the orthography where always the voiced consonant is kept.

$$[1] \quad C_x:C_y \quad \alpha _ \# : 0 \quad ;$$

where C_x in (b d g)

C_y in (p t k) matched;

```
#lõ~b# #rã~d# #we:g# #we:g+e#
||| 1| ||| 1| ||| 1| ||| |||
0lõ~p0 0rã~t0 0we:k0 0we:g0e0
```

The two-level rule realises b, d and g as their voiceless counterparts p, t, and k respectively whenever directly followed by a boundary.

While the original linguistic motivation behind two-level morphology was SPE and two-level rules were designed to describe morphophonology the mechanism can deal with a much wider range of phenomena.

4.3.3 Umlaut in German

German umlaut is used to mark--among other morphosyntactic features--plural.

pili Æ pinipili tahi Æ tinatahi

```
#X+R00E+pili# #X+R00E+tahi#  
|1|2||3||| |1|2||3|||  
000pini0pili0 000tina0tahi0
```

5 Further reading and relevant resources

The most comprehensive book about computational morphology is Richard Sproat's book *Morphology and Computation* (Sproat 1992). It gives a concise introduction into morphology with examples from various languages and a good overview of applications of computational linguistics. On the methodological side it concentrates on finite-state morphology omitting other paradigms. *Computational Morphology* (Black et al. 1992) gives a more in-depth description of finite-state morphology but concentrates exclusively on English. An excellent overview of morphology with examples from diverse languages is found in the *Handbook of Morphology* (Spencer and Zwicky 1998).

To get some hands-on experience with morphological processing connect to [RXRC Europe](#) and [Lingsoft](#). A free downloadable version of a two-level morphology is available from [SIL](#).

References

1. Beesley K.R. and Karttunen L. 2000. *Finite-State Morphology: Xerox Tools and Techniques*. Cambridge University Press, Cambridge.
2. Black A.W., Ritchie G.D., Pulman S.G., Russell G.J. 1987. Formalisms for Morphographemic Description, Proc. 3rd European ACL, pp11-18, Copenhagen.
3. Calder J. 1989. Paradigmatic Morphology, Proc. 4th European ACL, pp58-65, Manchester.
4. Chomsky N. and Halle M.: *The Sound Pattern of English*, Harper & Row, Hagerstown/London/New York, 1968.
5. Evans R., Gazdar G. 1996 DATR.: A Language for Lexical Knowledge Representation, *Computational Linguistics* 22(2)167-216.
6. Fredkin E. 1960. Trie Memory, *Communications ACM* 3, pp490-499.
7. Fromkin V., Rodman R. 1983. *An Introduction to Language*. Holt, Rinehart & Winston, New York.
8. Kay M. 1987. Noncatenative finite-state morphology, in Proc. of the 3rd Conference of the European Chapter of the ACL, Copenhagen, Denmark, pp2-10.
9. Kiparsky P. 1987. The Phonology of Reduplication. Manuscript. Stanford University.
10. Kiraz G.A. 1996. SEMHE: A Generalized Two-Level System, in Proc. of 34th Annual Meeting of the Association for Computational Linguistics, Morgan Kaufmann, Los Altos, pp159-166.
11. Kiraz G.A. 1997. Compiling Regular Formalisms with Rule Features into Finite-State Automata, in Cohen P.R., Wahlster W.(eds.), Proc. 35th Annual Meeting of the Association for Computational Linguistics, Morgan Kaufmann, Los Altos, pp329-336.
12. Koskenniemi K. 1984. A General Computational Model for Word-Form Recognition and Production, Proceedings 10th International Conference on Computational Linguistics, Stanford, CA.
13. Nash D. 1980. Topics in Warlpiri Grammar, PhD Thesis, MIT, Cambridge, MA.
14. Nida E. 1949. *Morphology: The Descriptive Analysis of Words*. University of Michigan Press.
15. Ritchie G.D., Russel G.J., Black A.W., Pulman S.G. 1991. *Computational Morphology*, MIT Press, Cambridge.
16. Rose S. 2000. Triple Take: Tigre and the case of internal reduplication. *Studies in Afroasiatic Grammar*.

17. Ruessink H. 1989. Two-Level Formalisms, Working Papers in Natural Language Processing 5, Rijksuniversiteit Utrecht.
18. Spencer A., Zwicky A. (eds.) 1998. The Handbook of Morphology, Basil Blackwell, Oxford.
19. Sproat R.W., 1992. *Morphology and Computation*, MIT Press, Cambridge, MA.
20. Trost H. 1992. X2MORPH: A Morphological Component Based on Augmented Two-Level Morphology, in Proc. 12th International Joint Conference on Artificial Intelligence, Sydney, Morgan Kaufmann, San Mateo, pp.1024-1030.
21. Trost H. 1993. Coping With Derivation in a Morphological Component, in Proc. 6th Conference of the European Chapter of the Association for Computational Linguistics, Utrecht, pp368-376.