Chapter NLP:IV

IV. Words

- Morphology
- Word Classes
- Named Entities

Morphology

Overview [Hancox 1996]

Morphology is the study of the structure and formation of words.

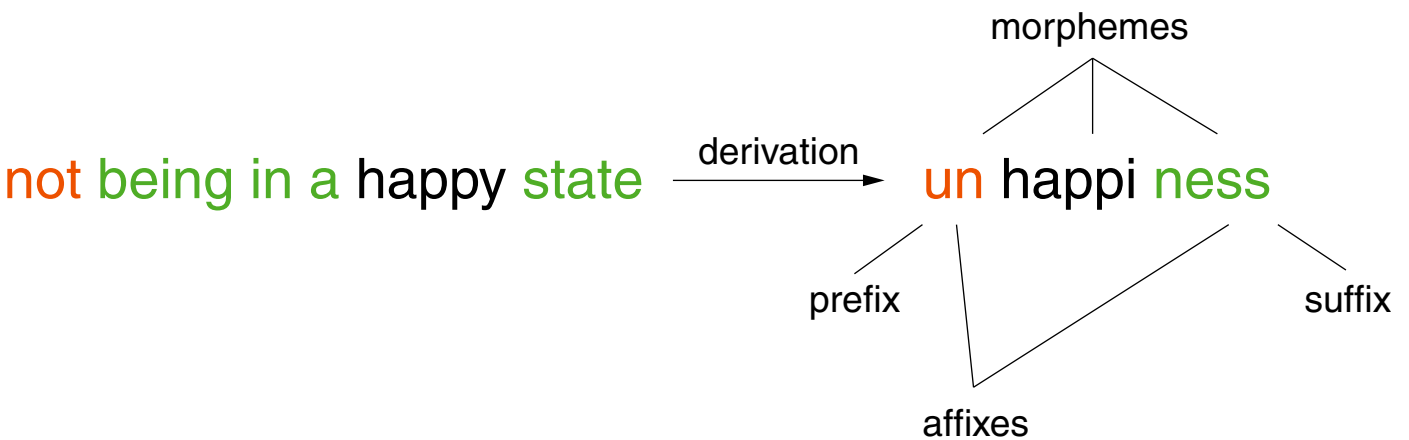
call in the past $\xrightarrow{\text{inflection}}$ call ed

not being in a happy state $\xrightarrow{\text{derivation}}$ un happi ness

Morphology

Overview [Hancox 1996]

Morphology is the study of the structure and formation of words.

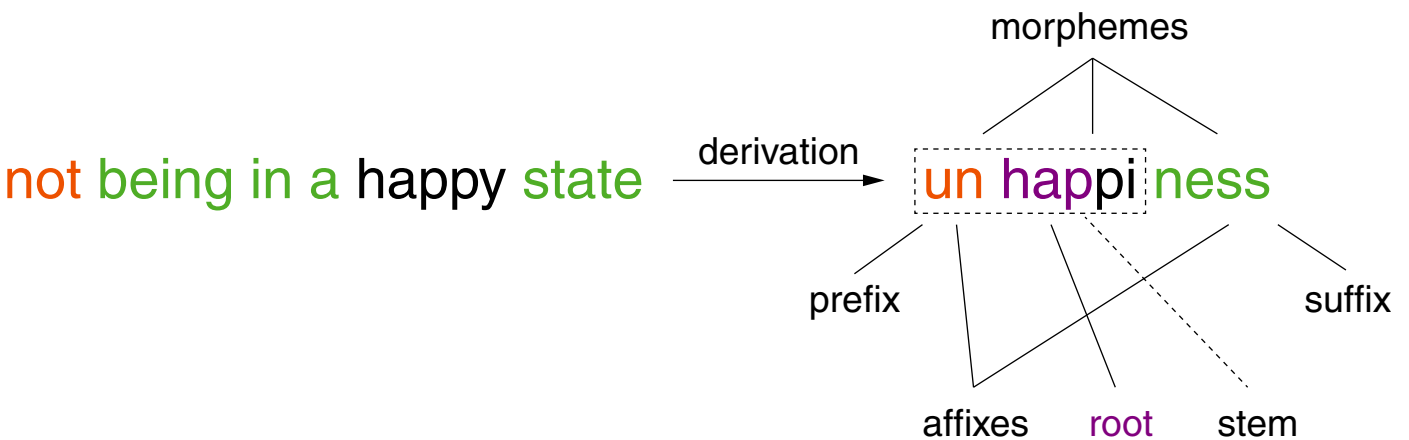


- A morpheme is a “minimal unit of meaning”.
  Free morphemes can also be used as words.
  Bounded morphemes appear only as affixes (prefix, suffix, infix, and more) to words.

Morphology

Overview [Hancox 1996]

Morphology is the study of the structure and formation of words.

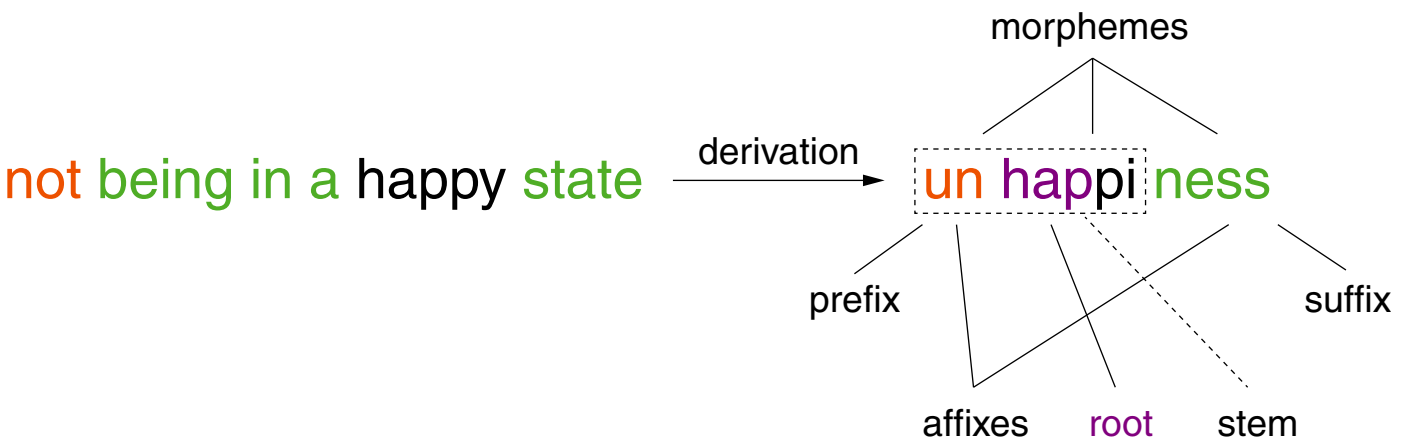


- A morpheme is a “minimal unit of meaning”.
  Free morphemes can also be used as words.
  Bounded morphemes appear only as affixes (prefix, suffix, infix, and more) to words.
- A root is a single morpheme, a stem one or more.
  A root is the derivational base, or type, of a word, a stem its inflectional base.

Morphology

Overview [Hancox 1996]

Morphology is the study of the structure and formation of words.



- A morpheme is a “minimal unit of meaning”.
  Free morphemes can also be used as words.
  Bounded morphemes appear only as affixes (prefix, suffix, infix, and more) to words.
- A root is a single morpheme, a stem one or more.
  A root is the derivational base, or type, of a word, a stem its inflectional base.

➜ Morphological analysis: identification of a word’s morphemes and their role.

Morphology

Stemming

Mapping of a word token to its word stem by removal of inflection (e.g., affixes).

Inflections:

- noun declination (grammatical case, numerus, gender)
- verb conjugation (grammatical person, numerus, tense, mode, . . .)
- adjective and adverb comparison

Example:

connect	connects
	connected
	connecting
	connection

Morphology

Stemming: Principles [Frakes 1992]

1. Table lookup:
   Given a word stem, store its inflections in a hash table. Problem: completeness.

2. Affix elimination:
   Rule-based algorithms to identify prefixes and suffixes. Given their efficiency and intuitive workings, these are most commonly used.

3. Character n-grams:
   Usage of 4-grams or 5-grams from tokens as stems. Basic heuristic for English: use the first 4 characters as stem.

4. Successor variety:
   Exploits knowledge about structural linguistics to identify morpheme boundaries. The character sequences of tokens are added to a trie data structure; the outdegrees of inner nodes are analyzed to find suitable stems. Problem: difficult to operationalize.

Morphology

Stemming: Affix Elimination

Idea: If the word ends in a known suffix, remove the suffix (iteratively).

1. Remove the longest possible match based on a set of rules.
2. Repeat Step 1 until no rule can be applied anymore.
3. Re-code to address irregularities captured by the rules.

Morphology

Stemming: Porter Stemmer

The Porter Stemmer is an implementation of affix elimination with rules.

- 9 rule sets, each consisting of 1-20 rules.
- Rules of each group are sorted, to be applied top to bottom.
- Move through the rule sets and apply the first match (or none).
- Rules are defined as follows: `<Premise> S1` $\longrightarrow$ `S2`.

If a word to be stemmed ends with `S1` and if the subsequence before `S1` (= word stem) fulfills the `<Premise>`, replace `S1` by `S2`

Morphology

Stemming: Porter Stemmer

The Porter Stemmer is an implementation of affix elimination with rules.

- 9 rule sets, each consisting of 1-20 rules.
- Rules of each group are sorted, to be applied top to bottom.
- Move through the rule sets and apply the first match (or none).
- Rules are defined as follows: `<Premise> S1` $\longrightarrow$ `S2`.

If a word to be stemmed ends with `S1` and if the subsequence before `S1` (= word stem) fulfills the `<Premise>`, replace `S1` by `S2`

Premises:

`(m>x)`	Number of vowel-consonant-sequences is larger than `x`.
`(*S)`	Word stem ends with `S`.
`(*v*)`	Word stem contains a vowel.
`(*o)`	Word stem ends with `cvc`, where the second consonant `c` $\notin$ `{W, X, Y}`.
`(*d)`	Word stem ends with two identical consonants.

Morphology

Remarks:

Notation:

- `c` denotes a consonant, `C` a non-empty sequence of consonants.
  `v` denotes a vowel, `V` a non-empty sequence of vowels.

  $\rightarrow$ Every word is defined by $[\texttt{C}](\texttt{VC})^{\texttt{m}}[\texttt{V}]$

- Vowel: Letters `A`, `E`, `I`, `O`, and `U` as well as `Y` after a consonant.
  Example: In `toy` the `y` is a consonant, in `lovely` a vowel.

Morphology

Stemming: Porter Stemmer

Selection of rules:

Rule set	Premise	Suffix	Replacement	Example
1a	Null	sses	ss	caresses → caress
1a	Null	ies	i	ponies → poni
1b	(m>0)	eed	ee	agreed → agree feed → feed
1b	(*v*)	ed	ε	plastered → plaster bled → bled
1b	(*v*)	ing	ε	motoring → motor sing → sing
1c	(*v*)	y	i	happy → happi sky → sky
2	(m>0)	biliti	ble	sensibiliti → sensible

Morphology

Stemming: Porter Stemmer

Example:

```
Alan Mathison Turing was an English mathematician, computer scientist,
logician, cryptanalyst, philosopher, and theoretical biologist.  Turing
was highly influential in the development of theoretical computer
science, providing a formalisation of the concepts of algorithm and
computation with the Turing machine, which can be considered a model of a
general-purpose computer.  Turing is widely considered to be the father
of theoretical computer science and artificial intelligence.
```

Morphology

Stemming: Porter Stemmer

Example:

Alan Mathison Turing was an English mathematician, computer scientist, logician, cryptanalyst, philosopher, and theoretical biologist. Turing was highly influential in the development of theoretical computer science, providing a formalisation of the concepts of algorithm and computation with the Turing machine, which can be considered a model of a general-purpose computer. Turing is widely considered to be the father of theoretical computer science and artificial intelligence.

Morphology

Stemming: Porter Stemmer

Example:

```
Alan Mathison Ture  wa  an English mathematician, comput  scientist,
logician, cryptanalyst, philosoph  , and theoret    biologist.  Ture
wa  highli influenti  in the develop    of theoret    comput
scienc , provid   a formalis     of the concepts of algorithm and
computat   with the Ture  machin , which can be consid    a model of a
gener  -purpos  comput  .  Ture  is wide  consid    to be the father
of theoret    comput  scienc  and artifici  intellig   .
```

Morphology

Stemming: Porter Stemmer

Weaknesses of the algorithm:

- Difficult to modify. The effects of changes are hardly predictable.
- Tends to overgeneralize:
  `university/universe, organization/organ`
- Does not capture clear generalizations:
  `European/Europe, matrices/matrix, machine/machineri`

Morphology

Stemming: Krovetz Stemmer

The Krovetz stemmer combines a dictionary-based approach with rules:

1. Word looked up in dictionary
2. If present, replaced with word stem
3. If not present, word is checked for removable inflection suffixes
4. After removal, dictionary is checked again
5. If still not present, different suffixes are tried

Observations:

- Captures irregular cases such as `is, be, was`.
- Produces words not stems (more readable, similar to lemmatization)
- Comparable effectiveness to Porter stemmer
- Lower false positive rate, somewhat higher false negative rate

Morphology

Stemming: Stemmer Comparison

Porter Stemmer:

```
Alan Mathison Turing was an English mathematician, computer scientist,
logician, cryptanalyst, philosopher, and theoretical biologist.  Turing
was highly influential in the development of theoretical computer
science, providing a formalisation of the concepts of algorithm and
computation with the Turing machine, which can be considered a model of a
general-purpose computer.  Turing is widely considered to be the father
of theoretical computer science and artificial intelligence.
```

Krovetz Stemmer:

```
Alan Mathison Turing was an English mathematician, computer scientist,
logician, cryptanalyst, philosopher, and theoretical biologist.  Turing
was highly influential in the development of theoretical computer
science, providing a formalisation of the concepts of algorithm and
computation with the Turing machine, which can be considered a model of a
general-purpose computer.  Turing is widely considered to be the father
of theoretical computer science and artificial intelligence.
```

Morphology

Stemming: Stemmer Comparison

Porter Stemmer:

Alan Mathison Turing was an English mathematician, computer scientist, logician, cryptanalyst, philosopher, and theoretical biologist. Turing was highly influential in the development of theoretical computer science, providing a formalisation of the concepts of algorithm and computation with the Turing machine, which can be considered a model of a general-purpose computer. Turing is widely considered to be the father of theoretical computer science and artificial intelligence.

Krovetz Stemmer:

Alan Mathison Turing was an English mathematician, computer scientist, logician, cryptanalyst, philosopher, and theoretical biologist. Turing was highly influential in the development of theoretical computer science, providing a formalisation of the concepts of algorithm and computation with the Turing machine, which can be considered a model of a general-purpose computer. Turing is widely considered to be the father of theoretical computer science and artificial intelligence.

Morphology

Stemming: Stemmer Comparison

Porter Stemmer:

```
Alan Mathison Ture  wa  an English mathematician, comput  scientist,
logician, cryptanalyst, philosoph  , and theoret  biologist.  Ture
wa  highli influenti  in the develop  of theoret  comput
scienc , provid  a formalis  of the concepts of algorithm and
computat  with the Ture  machin , which can be consid  a model of a
gener -purpos comput  .  Ture  is wide  consid  to be the father
of theoret  comput  scienc  and artifici  intellig  .
```

Krovetz Stemmer:

```
Alan Mathison Ture  was an English mathematician, computer scientist,
logician, cryptanalyst, philosopher, and theoretical biologist.  Ture
was highly influential in the development of theoretical computer
science, providing a formalise  of the concepts of algorithm and
computation with the Ture  machine, which can be considered a model of a
general-purpose computer.  Ture  is widely considered to be the father
of theoretical computer science and artificial intelligence.
```

Morphology

Stemming: Character n-grams [McNamee et al. 2004] [McNamee et al. 2008]

A substring of length n from a longer string is called a character n-gram. A string of length $m \geq n$ has at most $(m - n) + 1$ character n-grams.

Example: `Alan Mathison Turing ...`

- 1-grams: `A, l, a, n, M, a, t, h, i, s, o, n, T, u, r, i, n, g`
- 2-grams: `Al, la, an, Ma, at, th, hi, is, so, on, Tu, ur, ri, in, ng`
- 3-grams: `Ala, lan, Mat, ath, thi, his, iso, son, Tur, uri, rin, ing`
- 4-grams: `Alan, Math, athi, this, hiso, ison, Turi, urin, ring`
- 5-grams: `Alan, Mathi, athis, thiso, hison, Turin, uring`

Morphology

Stemming: Character n-grams [McNamee et al. 2004] [McNamee et al. 2008]

A substring of length n from a longer string is called a character n-gram. A string of length $m \geq n$ has at most $(m - n) + 1$ character n-grams.

Example: `Alan Mathison Turing ...`

- 1-grams: `A, l, a, n, M, a, t, h, i, s, o, n, T, u, r, i, n, g`
- 2-grams: `Al, la, an, Ma, at, th, hi, is, so, on, Tu, ur, ri, in, ng`
- 3-grams: `Ala, lan, Mat, ath, thi, his, iso, son, Tur, uri, rin, ing`
- 4-grams: `Alan, Math, athi, this, hiso, ison, Turi, urin, ring`
- 5-grams: `Alan, Mathi, athis, thiso, hison, Turin, uring`

Use the first (or all) character n-grams for $n = 4$ or $n = 5$ as pseudo-stems of a word.

Observations:

- Language-independent; good performance for many languages.
- Well-developed stemmers yield better performance (e.g., for English).
- Large overhead in terms of vocabulary size.

Morphology

Lemmatization

Problems with stemming:

- overstemming: artificial ambiguity

  `{organization, organ}` → `organ`

- understemming: unification fails

  `European` → `european`, `Europe` → `europ`

Idea: Look up canonical form of a word (lemmatization) from a dictionary:

inflected_type	**lemma_type**
European	Europe
Europe	Europe
Organizations	Organization

Problems with lookup approaches:

- Creating a good resoruces is labour intensive and error prone.
- Lookup lists will be incomplete / outdated quickly. consider spelling mistakes
- Lookup from long lists needs a lot of compute.