

Structural Linguistics 1: the *form*

Mariana Romanyshyn,
Computational Linguist at Grammarly

Contents

A word is its...

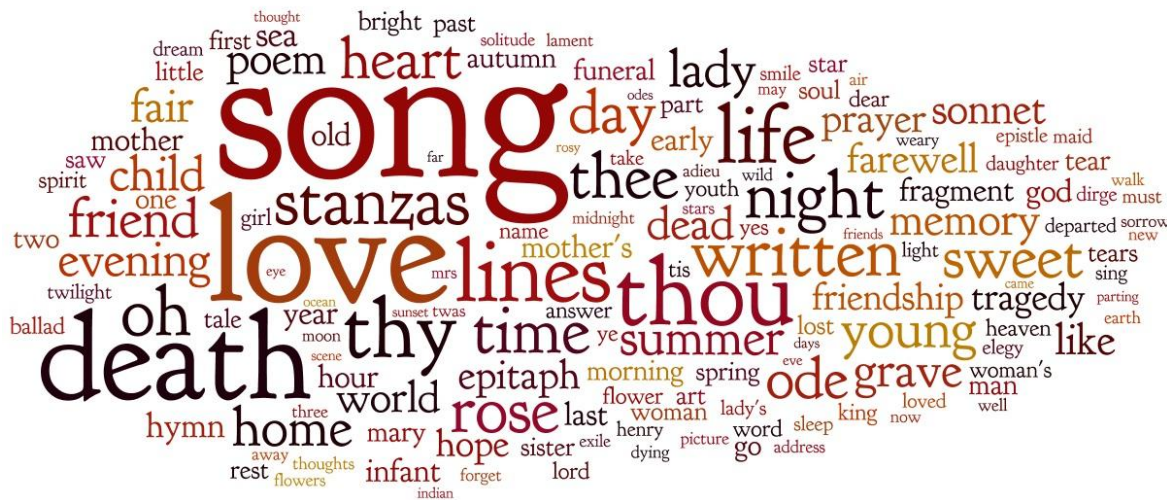
1. form
2. function
3. meaning



Contents

A word is its...

1. **form**
2. function
3. meaning



1. Orthography

Orthography

Investigate the way the word is written:

- capitalization
- hyphen
- apostrophe



Capitalization

- Style
 - sentence start
 - book titles
- Proper names
 - *Cities like **The Hague**...* (en)
 - *Учора під **Верховною Радою Саакашвілі**...* (uk)
 - *Мату в **Polsce** wybudowany dom...* (pl)

Capitalization

- Common nouns and adjectives
 - *On **Monday**, an **English Democrat**...* (en)
 - *Eine **Katze** kommt ins **Haus**.* (de)
- Nouns in the legal language
 - *... а **Замовник** має право...* (uk)
 - *The **Landlord** agrees to let the **Tenant** take...* (en)



Capitalization

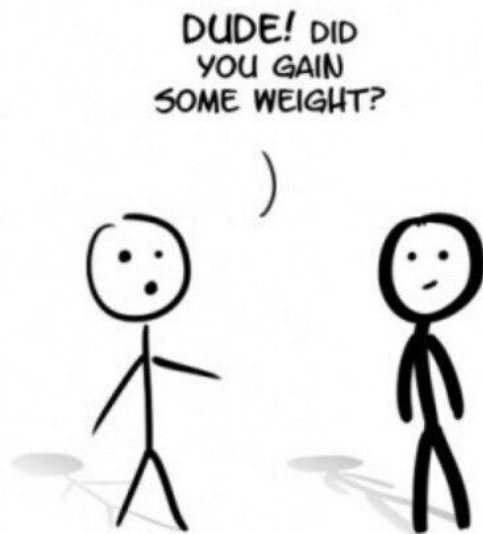
- Pronouns
 - **Sie** (de), **Вы** (ru), **Шумо** (tg), **Vi** (sl), *risponder***Le** (it)
 - *follow* **His** word (en), вчення **Його** сина (uk)
- Acronyms
 - **EU** (en), **ЄС** (uk), **UE** (pl)
 - **LotR** (en), **GmbH** (de)



Capitalization

Capitonyms - words that change meaning due to capitalization.

- **Laut** vs. **laut** (de)
- **Morgen** vs. **morgen** (de)
- **March** vs. **march** (en)
- **Polish** vs. **polish** (en)
- **China** vs. **china** (en)
- **Роман** vs. **роман** (uk)
- **Рада** vs. **рада** (uk)
- **Peru** vs. **peru** (pt)



Hyphenation

- Compounds
 - ***Editor-in-Chief*** (en), ***Ростов-на-Дону*** (ru)
 - ***passer-by*** (en), ***механіко-математичний*** (uk)
 - ***a twenty-five-year-old*** woman (en)
 - ***a please-don't-ask-me*** attitude (en)
- Prefixation/Suffixation
 - ***co-worker*** (en), ***екс-чемпіон*** (uk), ***celui-ci*** (fr)

Hyphenation

- Suspense
 - *pre- and post-operative* (en)
 - *радіо- і телепрограми* (uk)
- Onomatopoeia
 - *heh-heh* (en), *ціп-ціп-ціп* (uk), *fiu-fiu* (pl)



Hyphenation

Hyphenation eliminates ambiguity:

- *unionized* vs. *un-ionized*



Hyphenation

Hyphenation eliminates ambiguity:

- ***unionized*** vs. ***un-ionized***
- ***three-hundred-year-old*** trees

vs.

*three **hundred-year-old** trees*

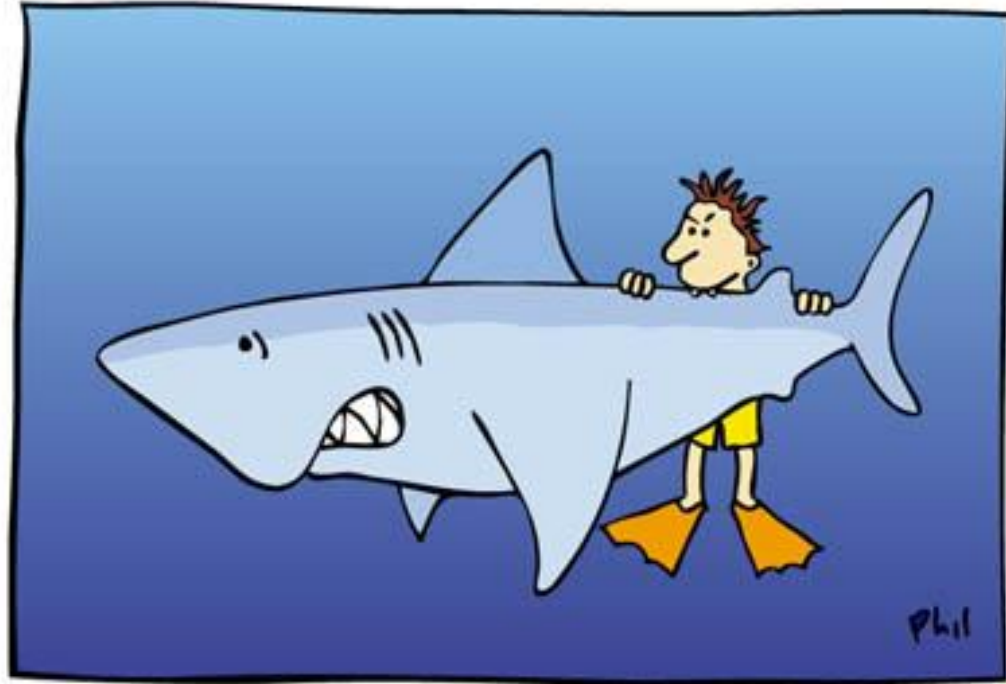
vs.

*three hundred **year-old** trees*



Hyphenation

a man-eating shark vs. *a man eating shark*

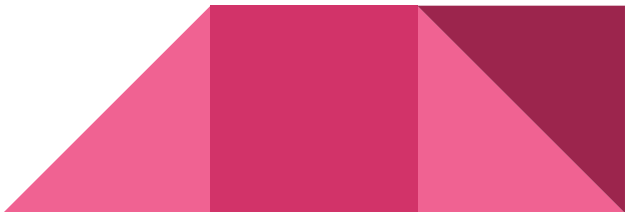


Apostrophes

- Contractions
 - *we'll, I'd've, 'cause, talkin'* (en)
 - *Wie geht's?* (de), *c'est* (fr), *l'opera* (it)
- Possession
 - *lady's, ladies'* (en)
- Names
 - *O'Doole, M'Gregor*
- Pronunciation
 - *бур'ян* (uk), *ка'а* (gn)

Usage in NLP

Capitalization:

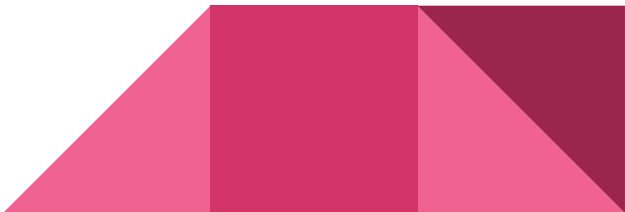
- Named entity recognition
 - Part-of-speech tagging
 - Truecasing
 - Title identification/formatting
 - Detection of politeness level
 - Word sense disambiguation
 - Error correction
- 

Usage in NLP

Hyphenation:

- Tokenization
- Part-of-speech tagging

Apostrophes:

- Tokenization
 - Formality detection and transfer
 - Text to speech, speech to text
- 

2. Tokenization

Tokenization

A token is:

- an independent word
- a number
- a punctuation mark



Tokenization

How many tokens are there in...

- *can't, we'll, Homer's, l'opéra, geht's*
- *ladies', 'cause, 'cause'*



Tokenization

How many tokens are there in...

- *can't, we'll, Homer's, l'opera, geht's*
- *ladies', 'cause, 'cause'*
- *twenty-five-year-old, жар-птиця, йди-но, такий-от*



Tokenization


How many tokens are there in...

- *can't, we'll, Homer's, l'opera, geht's*
- *ladies', 'cause, 'cause'*
- *twenty-five-year-old, жар-птиця, йди-но, такий-от*
- *gonna, lemme, dunno, sorta*




Tokenization

How many tokens are there in...

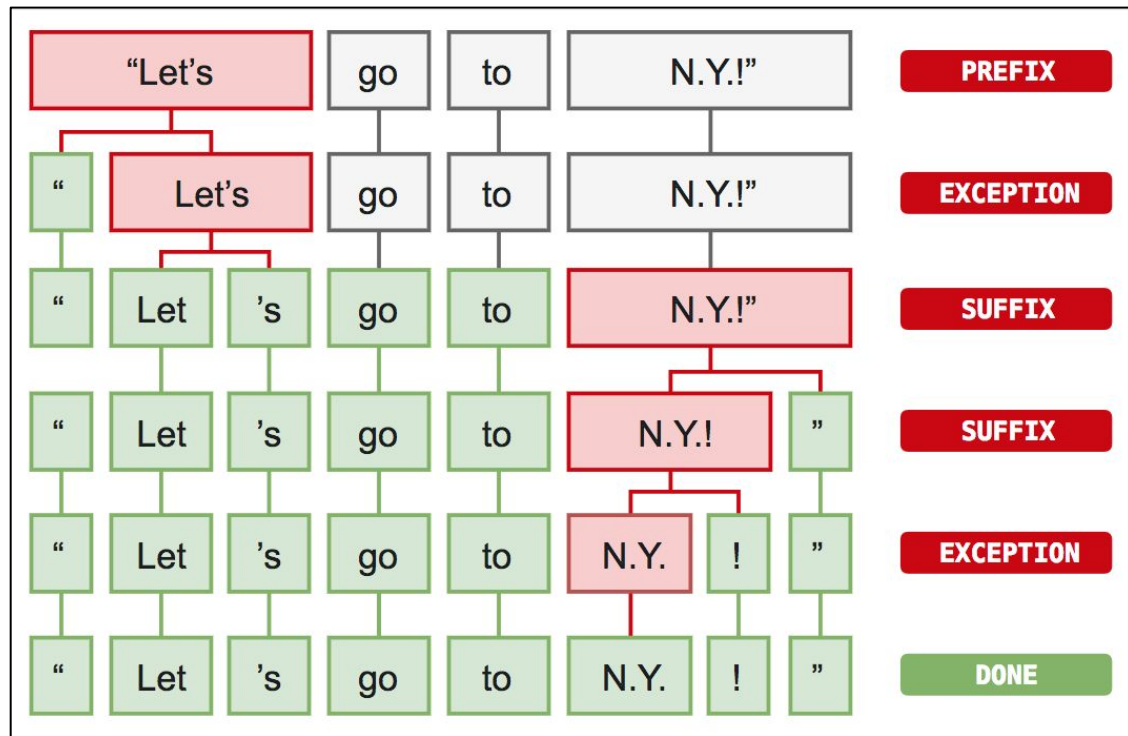
- *can't, we'll, Homer's, l'opera, geht's*
 - *ladies', 'cause, 'cause'*
 - *twenty-five-year-old, жар-птиця, йди-но, такий-от*
 - *gonna, lemme, dunno, sorta*
 - *\$3b, 1.35, 1/2/2018*
 - *etc., U.S.A.*
 - *:), www.example.com*
- 

Tokenization

How many tokens are there in...


- *can't, we'll, Homer's, l'opera, geht's*
 - *ladies', 'cause, 'cause'*
 - *twenty-five-year-old, жар-птиця, йди-но, такий-от*
 - *gonna, lemme, dunno, sorta*
 - *\$3b, 1.35, 1/2/2018*
 - *etc., U.S.A.*
 - *:), www.example.com*
 - *San Francisco, out of*
- 

Tokenization



Tokenization

How to do it?

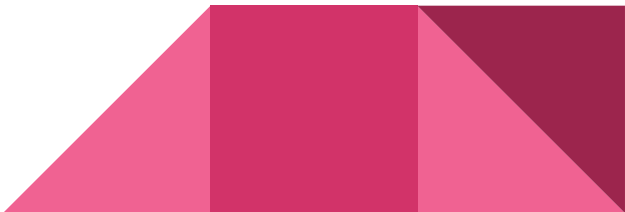
- [Stanford CoreNLP](#) (6 languages; wrappers for [10+ programming langs](#))
 - [Spacy](#) (7 languages; Python)
 - [OpenNLP](#) (7 languages; Java)
 - [Emory NLP](#) (English; Java)
 - [lang-uk](#) (Ukrainian; Python)
 - [languagetool](#) (30 languages; Java), or [nlp_uk](#) (Ukrainian; Groovy)
 - [cl-nlp](#) (Common Lisp)
 - [nltk](#) (English; Python) or [TextBlob](#) (English; Python)
 - Write your own rules and regexps :)
- 

Tokenization in spaCy

```
In [1]: import en_core_web_md  
  
nlp = en_core_web_md.load()
```

```
In [2]: quote = nlp("Roads? Where we're going we don't need roads.")  
" ".join([token.text for token in quote])
```

```
Out[2]: "Roads ? Where we 're going we do n't need roads ."
```



3. Morphology

Morphemes

A word consists of morphemes:

- stem+
- affix*



Morphemes

A word consists of morphemes:

- stem+
 - **board, blackboard** (en)
 - **рука, рукопис** (uk)
 - **Lehrer, Lehrbuch** (de)
- affix*
 - **a**board, blackboards**s** (en)
 - **підручний, рукописний** (uk)
 - Lehr**er**in, Lehrbüch**er** (de)



Affixes

Affixes by position:

- prefix - **over**estimate, **під**купити
- suffix/postfix - overestimation**, підкупитися**
- interfix - speed**o**meter, руко**о**пис
- circumfix/confix - **gesproche**n, **до**чека**ти**ся
- infix - abso-**bloody**-lutely, спат**онь**ки



Affixes-shmaffixes

- Duplifix :)
 - *hokey-pokey, teenie-weenie*
 - *article-shmarticle, fancy-shmancy*
 - *helter-skelter*
 - *kitty-cat, chit-chat*
 - *bye-bye, choo-choo*



Affixes

Affixes by function:

- inflectional
- derivational



Affixes

Affixes by function:

- inflectional
 - *fox* => *fox**es***, *стіл* => *столи*, *Tasse* => *Tassen*
 - *high* => *high**er***, *красивий* => *найкрасиві**ш**ий*
 - *smell* => *smell**ing***, *нюхати* => *нюхаю*, *брати* => *забрати*
- derivational
 - *schön* => *Schön**heit***, *краса* => *краси**в**ий*
 - *happy* => *happi**ness***, *хмара* => *безхмар**н**ий*
 - *sprechen* => *abgesprochene*

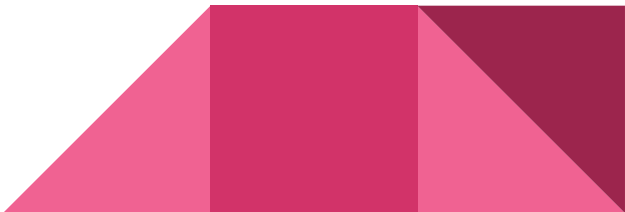
Inflectional Affixes

- change the word form of a lexeme
- keep the part of speech
- examples:
 - *чита**в***, *чита**ти**му*, *прочита**в***, *чита**й*** => *чита**ти*** (verb)
 - *high**er***, *high**est*** => *high* (adj)



Synthetic vs. Analytic Languages

- Synthetic - relation between words is expressed by **inflection**
 - *Я тобі напишу. Напишу я тобі. Тобі я напишу.*
 - *Я напишу тобі. Напишу тобі я. Тобі напишу я.*
- Analytic language - relation between words is expressed by **word order** and **helper words**
 - *Ich werde dir schreiben. Dir werde ich schreiben.*
 - *I will write to you.*



Synthetic vs. Analytic Languages

- *the kin**est**, **most** rational* (en)
- *добр**ей**ший, **самый** добрый* (ru)
- *найдобр**і**ший, наиб**іль**ш добрий, ~~самий добрий~~* (uk)

- ***will** write* (en)
- ***буду** писать* (ru)
- ***буду** писати, писат**иму*** (uk)



Synthetic vs. Analytic Languages

- *the kin**dest**, **most** rational* (en)
- *добр**ей**ший, **самый** добрый* (ru)
- *найдобр**і**ший, **найбільш** добрий, ~~самий добрий~~* (uk)

- ***will** write* (en)
- ***буду** писать* (ru)
- ***буду** писати, писат**иму*** (uk)

- *Evinizdeyim.* (tr) - Я є у Вас вдома.
- *Evinizdeymişim.* (tr) - Я був у Вас вдома.

**DO NOT TOUCH MORPHEME
WITH BARE HANDS.**



RISK OF INFLECTION.

Affixes

Affixes by function:

- inflectional
 - *fox* => *fox**es***, *стіл* => *столи*, *Tasse* => *Tassen*
 - *high* => *high**er***, *красивий* => *найкрасиві**ш**ий*
 - *smell* => *smell**ing***, *нюхати* => *нюхаю*, *брати* => *забрати*
- derivational
 - *schön* => *Schön**heit***, *краса* => *краси**в**ий*
 - *happy* => *happi**ness***, *хмара* => *безхмар**н**ий*
 - *sprechen* => *abgespro**chen***

Derivational Affixes

- create a new lexeme, but keep the stem
- may change the part of speech
- examples:
 - *schön* => *Schön**heit*** (*schön*, adj => noun)
 - *хмара* => ***без**хмарний* (*хмар*, noun => adj)
 - *sprechen* => ***ab**gesprochen* (*sprech*, verb => adj)



Derivational Affixes

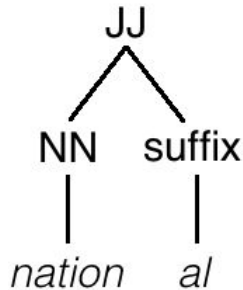
Word derivation: how to get from ***nation*** to ***denationalisation***?

NN
|
nation



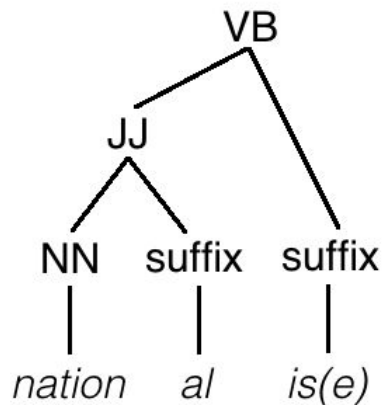
Derivational Affixes

Word derivation: how to get from *nation* to *denationalisation*?



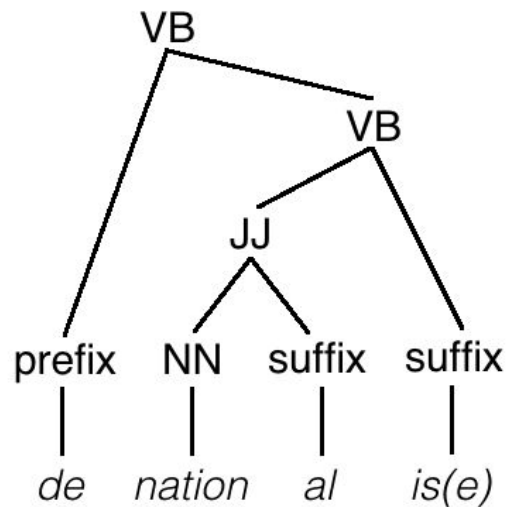
Derivational Affixes

Word derivation: how to get from *nation* to *denationalisation*?



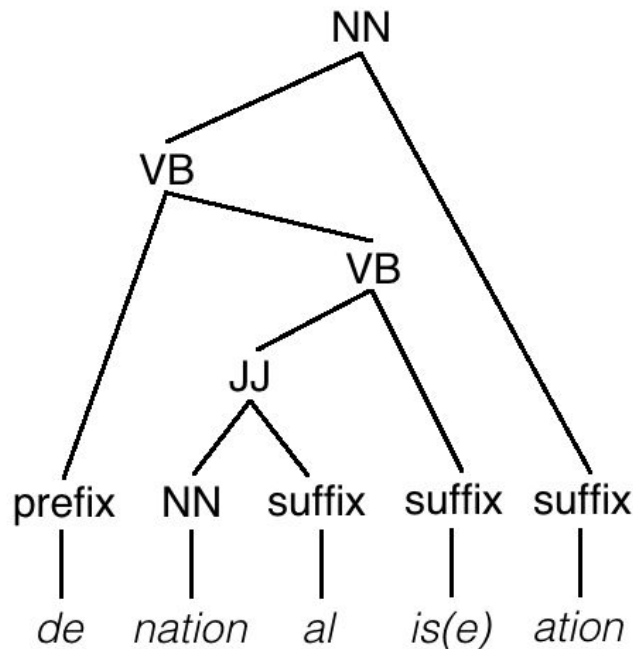
Derivational Affixes

Word derivation: how to get from *nation* to *denationalisation*?



Derivational Affixes

Word derivation: how to get from *nation* to *denationalisation*?



More exercise?

What is the derivation of...

- *counterproductiveness*
- *antirepresentationalist*
- *поназдоганяти*
- *перекотиполе*
- *учительствовать*
- *сверхприбыльный*



Affixes

Affixes by function:

- inflectional
 - change the word form of a lexeme
 - keep the part of speech
- derivational
 - create a new lexeme, but keep the stem
 - may change the part of speech



Affixes

Affixes by function:

- inflectional => lemmatization
- derivational => stemming

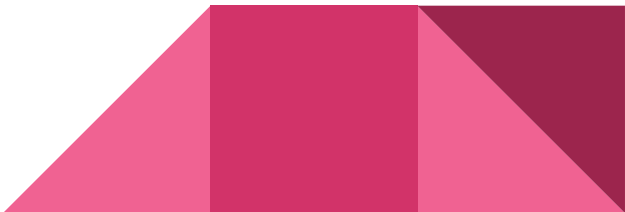


Lemmatization

Lemma - the base form of the word.

- *foxes* => *fox*
- *найкрасивіший* => *красивий*
- *smelling* => *smell*

How to lemmatize?

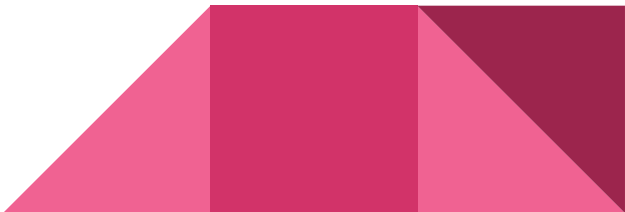
- NLP toolkits mentioned previously
 - [pymorphy2](#) (Russian, Ukrainian)
 - write your own lemmatizer :)
- 

Lemmatization in spaCy

```
In [3]: import en_core_web_md  
  
nlp = en_core_web_md.load()
```

```
In [4]: quote = nlp("Roads? Where we're going we don't need roads.")  
" ".join([token.lemma_ for token in quote])
```

```
Out[4]: 'road ? where -PRON- be go -PRON- do not need road .'
```



Lemmatization in spaCy

```
In [3]: import en_core_web_md  
  
nlp = en_core_web_md.load()
```

```
In [4]: quote = nlp("Roads? Where we're going we don't need roads.")  
" ".join([token.lemma_ for token in quote])
```

```
Out[4]: 'road ? where -PRON- be go -PRON- do not need road .'
```

```
In [5]: quote = nlp("You're gonna need a bigger boat.")  
" ".join([token.lemma_ for token in quote])
```

```
Out[5]: '-PRON- be go to need a big boat .'
```

Lemmatization in pymorphy2

```
In [1]: import tokenize_uk  
import pymorphy2
```

```
In [2]: text = "Основні етапи купівлі та розмитнення \"євробляхи\"."  
morph = pymorphy2.MorphAnalyzer(lang='uk')
```

```
In [3]: tokens = tokenize_uk.tokenize_words(text)  
print(tokens)
```

```
['Основні', 'етапи', 'купівлі', 'та', 'розмитнення', '"', 'євробляхи', '"', '.']
```

```
In [4]: print([morph.parse(word)[0].normal_form for word in tokens])
```

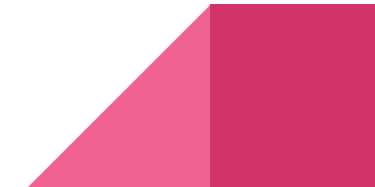
```
['основний', 'етап', 'купівля', 'та', 'розмитнення', '"', 'євробляха', '"', '.']
```

Stemming

Stem - the base morpheme of the word.

- *Schönheit* => *schön*
- *безхмарний* => *хмар*
- *abgesprochene* => *sprech*

How to stem?

- [SnowballStem](#) (English, Russian, and 16 more languages)
 - PorterStemmer in [nltk](#) (English)
 - [Ukr_stemming](#) (Ukrainian)
 - write your own stemmer :)
- 

Stemming in SnowballStem

```
In [7]: import en_core_web_md
import snowballstemmer

nlp = en_core_web_md.load()
stemmer = snowballstemmer.stemmer('english')
```

```
In [8]: quote = nlp("This is the beginning of a beautiful friendship.")
print(" ".join([token.lemma_ for token in quote]))
print(" ".join(stemmer.stemWords([token.text.lower()
                                for token in quote])))
```

this be the beginning of a beautiful friendship .
this is the begin of a beauti friendship .

Usage in NLP

Lemmatization & Stemming

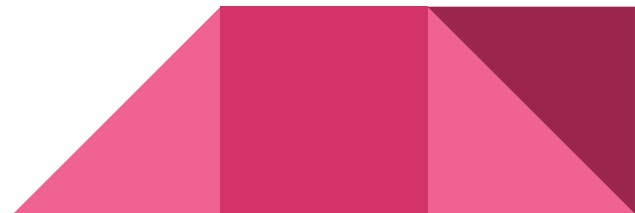
- text classification
- information retrieval
- dimensionality reduction

Lemmatization

- any task that needs a dictionary lookup

Stemming

- complex word identification



4. Other parts of the word

Word Parts

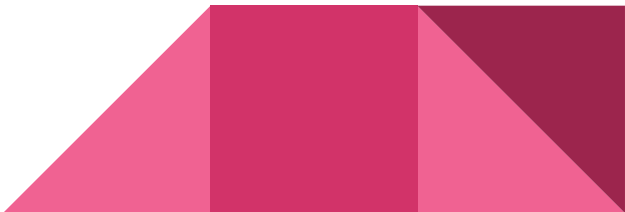
A word also consists of:

- characters
- syllables
- sounds/phonemes



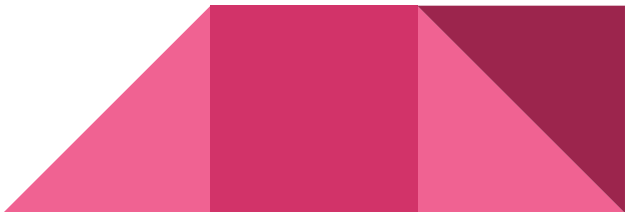
Usage in NLP

Characters, syllables, sounds:

- Text-to-speech, speech-to-text
 - Text readability/fluency
 - Text complexity
 - Spelling correction
 - Automatic hyphenation
 - Onomastics
- 

Conclusion

Features

- Capitalization, hyphenation, apostrophes
 - Number of tokens, position of a token
 - Lemma, stem
 - Number of stems
 - Number and types of affixes
 - Length of the word/lemma/stem
 - Number of syllables in a word
 - Ratio of vowels vs. consonants
 - Voiced vs. voiceless consonants
 - All possible frequencies
- 



Questions?