Linguistics in NLP: Why so complex?

Mariana Romanyshyn
Technical Lead, Computational Linguist at Grammarly

Contents

1. Motivation
2. Approach
3. Complex word identification
4. Complex word simplification
5. Conclusion

1. Motivation

Where do complex words come from?

Where do complex words come from?

Complex words come from complex texts!

What texts are complex?

Texts that are too complicated for non-specialists.

- Technical Medical Language
  - Hypertension risk factors include obesity,...
  - High blood pressure risk factors include excessive weight,...
- Legal Language
  - The Products transacted through the Service are...
  - The Products managed through the Service are...

Taken from https://slideplayer.com/slide/4453958/

What texts are complex?

Texts that are too complicated for:

- second language learners
- native speakers with low literacy levels
- people with reading impairments
- children

What texts are complex?

Or…

What texts are complex?

Or…

They are warm, nice people with big hearts.

What texts are complex?

Or…

They are warm, nice people with big hearts.

They are humid, prepossessing Homo Sapiens with full-sized aortic pumps.

Text simplification

Aim - to facilitate reading comprehension for

- non-specialists
- second language learners
- native speakers with low literacy levels
- people with reading impairments
- children

Text simplification

Aim - to facilitate reading comprehension for

- non-specialists
- second language learners
- native speakers with low literacy levels
- people with reading impairments
- children
- *other NLP applications*

Text simplification

Ways of simplification

- syntactic simplification
- lexical simplification
- explanation generation

Text simplification

Syntactic simplification:

London, which is the capital of the United Kingdom, is located in southeastern England.

London is the capital of the United Kingdom. It is located in southeastern England.

Text simplification

Lexical simplification:

They are humid, prepossessing Homo Sapiens with full-sized aortic pumps.

They are warm, nice people with big hearts.

Text simplification

Explanation generation:

The baby was born with pulmonary atresia.

The baby was born with pulmonary atresia.
Pulmonary atresia is a type of heart defect.

IBM Content Clarifier

Original content

Hey John, my family is in unanimous agreement about the IPhone being an astonishing invention. It was built under the helm of Steve Jobs, who was a masterful innovator. I bought mine from the Apple Store in New York City. Over the years, I have downloaded a humongous amount of apps from the App Store. Mary also owns an IPad if I'm not mistaken. By the way, if you're jealous, you really should replace your superannuated mobile phone!

Analyzed content

Hey John, my family is in solid agreement about the IPhone being an amazing invention. It was built under the direction of Steve Jobs, who was a skilled pioneer. I bought mine from the Apple Store in New York City. Over the years, I have downloaded a large amount of apps from the App Store. Mary also owns an IPad if I'm not mistaken. By the way, if you're jealous, you really should replace your old mobile phone!

Taken from https://www.ibm.com/blogs/

Grammarly

The patient was moribund when the doctor arrived.

Overly complex wording

It appears that *moribund* may not be the best word to use in this context. Consider replacing it with a more common synonym.

dying

MORE

IGNORE

2. Approach

What we already know

- two shared tasks on ***complex word identification*** (CWI) of [2016] and [2018]
- a separate CWI module helps
- traditional ML outperforms deep learning
- non-annotated data
  - Wikipedia and [Simple Wikipedia]
  - [Newsela]

The data isn't that good...

During this period , teams using Brabham cars won championships in Formula Two …

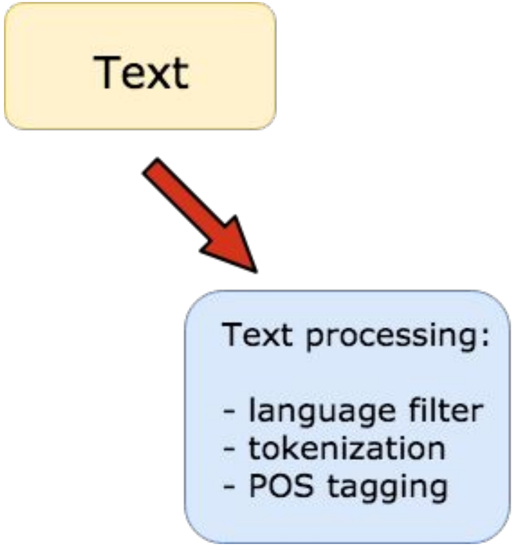
The energy is created by the laughter of the children when playing with the Boohbahs , the Boohball , and the Storypeople .

George Harrison described it in 1969 as `` one of those instant whistle-along tunes which some people hate , and other people really like .

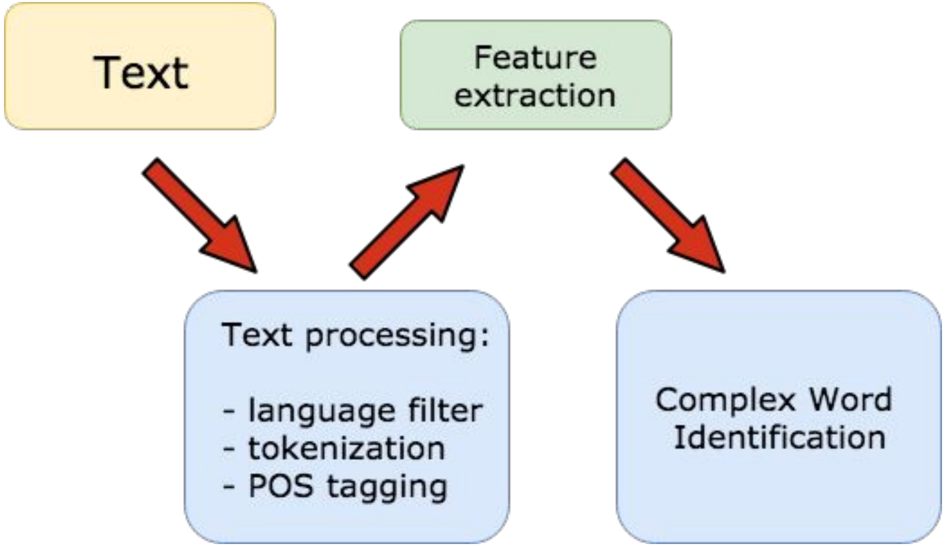
Pipeline

Text

Pipeline


Text
Text processing:
- language filter
- tokenization
- POS tagging

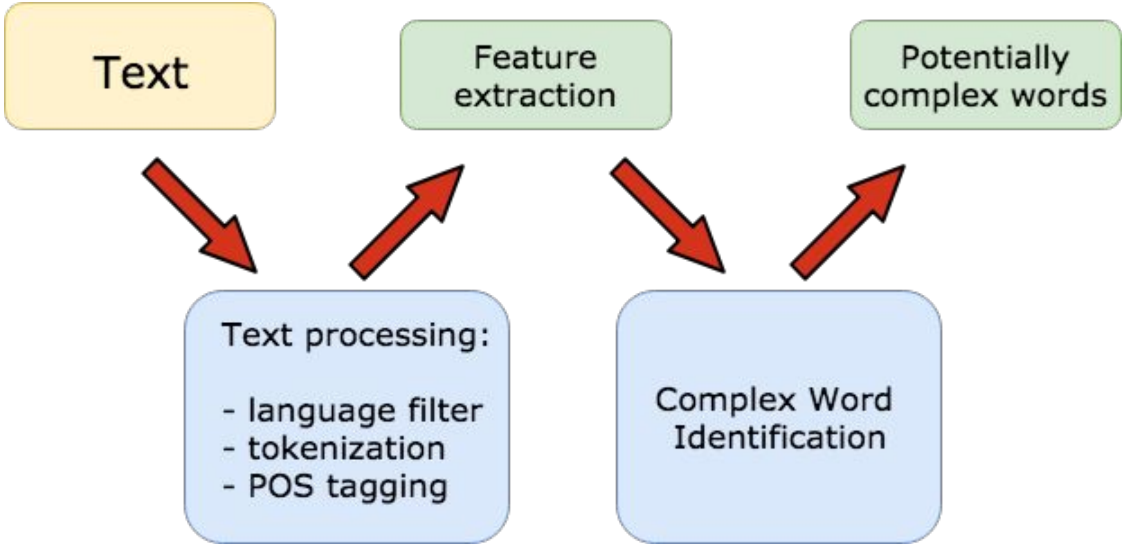
Pipeline



Features:

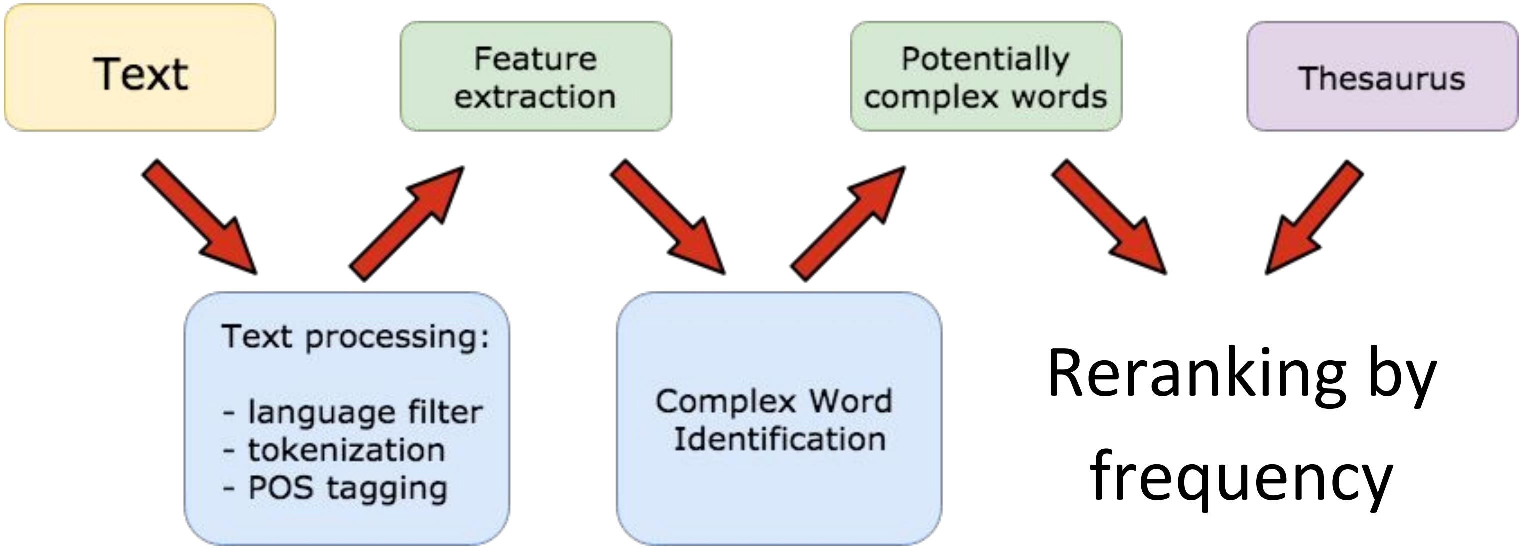
- word length
- word frequency
- part of speech

Pipeline

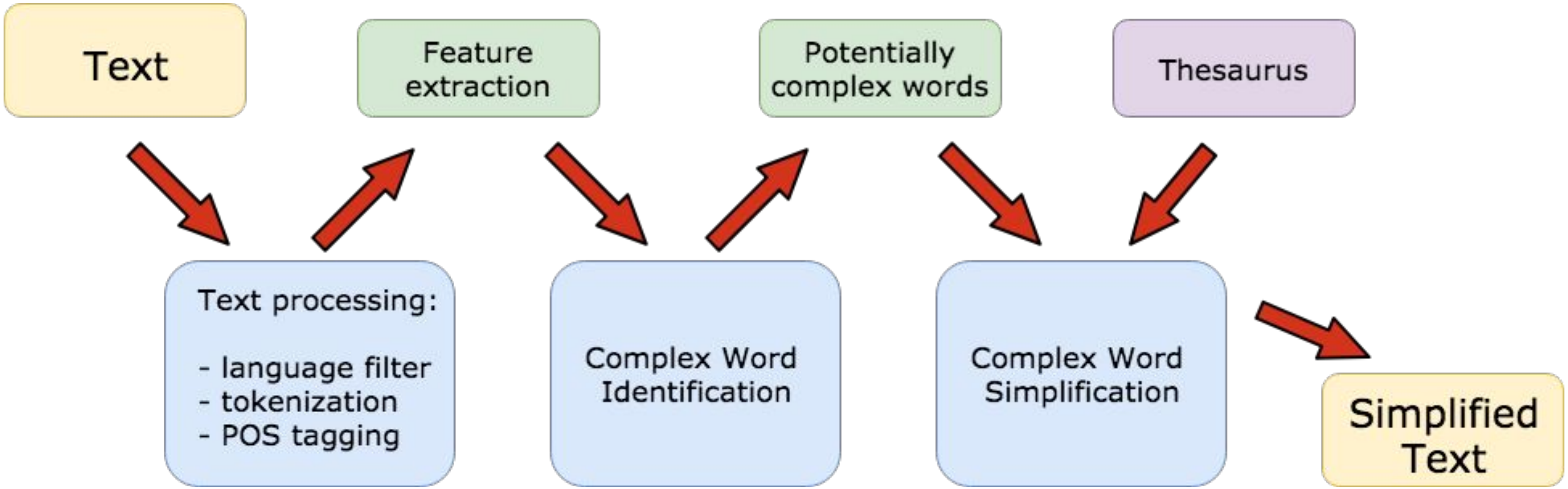


Text
Feature extraction
Potentially complex words
Text processing:
- language filter
- tokenization
- POS tagging
Complex Word Identification

Pipeline


Text
Feature extraction
Potentially complex words
Thesaurus
Text processing:
- language filter
- tokenization
- POS tagging
Complex Word Identification
Reranking by frequency

Pipeline


Text
Feature extraction
Potentially complex words
Thesaurus
Text processing:
- language filter
- tokenization
- POS tagging
Complex Word Identification
Complex Word Simplification
Simplified Text

How do we measure success?

How do we measure success?

Criteria of success ***by an NLP researcher***:

- good F-measure*
- OK speed

** on my test set*

How do we measure success?

Criteria of success ***by an NLP researcher***:

- good F-measure*
- OK speed

** on my test set*

One problem

NLP researcher is not the final consumer of the NLP application.

How do we measure success?

Criteria of success ***by an actual user***:

- consistent
- grammatically correct
- *indeed* simpler
- *not* too simple
- the meaning shouldn’t change


NOPE.
NOT GOOD ENOUGH
memegenerator.net

3. Complex word identification

1. Word Frequency

- get a large corpus
- tokenize it
- count

=> profit?

1. Word Frequency

Problem: inconsistency.

- *Ladies like to accessorize.*
- *That lady accessorizes her dress with a silver belt.*

1. Word Frequency

Problem: inconsistency.

- *Ladies like to accessorize.*
- *That lady accessorizes her dress with a silver belt.*

Why?

1. Word Frequency

Word means all forms of the word.

Freq = C(*"accessorize"*) + C(*"accessorizes"*) +

C(*"accessorized"*) + C(*"accessorizing"*)

1. Word Frequency

Word means all forms of the word.

Freq = C(*"accessorize"*) + C(*"accessorizes"*) +

C(*"accessorized"*) + C(*"accessorizing"*) +

C(*"accessorise"*) + C(*"accessorises"*) +

C(*"accessorised"*) + C(*"accessorising"*)

Inflectional Morphology

Many forms of the same word:

- *cute - cuter - cutest*
- *cat - cats*
- *do - does - did - done - doing*

Inflectional Morphology

Lemmatization:

- *cute -> cute, cuter -> cute, cutest -> cute*
- *cat -> cat, cats -> cat*
- *do -> do, does -> do, did -> do, done -> do, doing -> do*

2. Word Length

Problem: inconsistency.

- *You are a great friend.*
- *There was a climate of friendliness and cooperation in the team.*

2. Word Length

Some long words are actually simple:

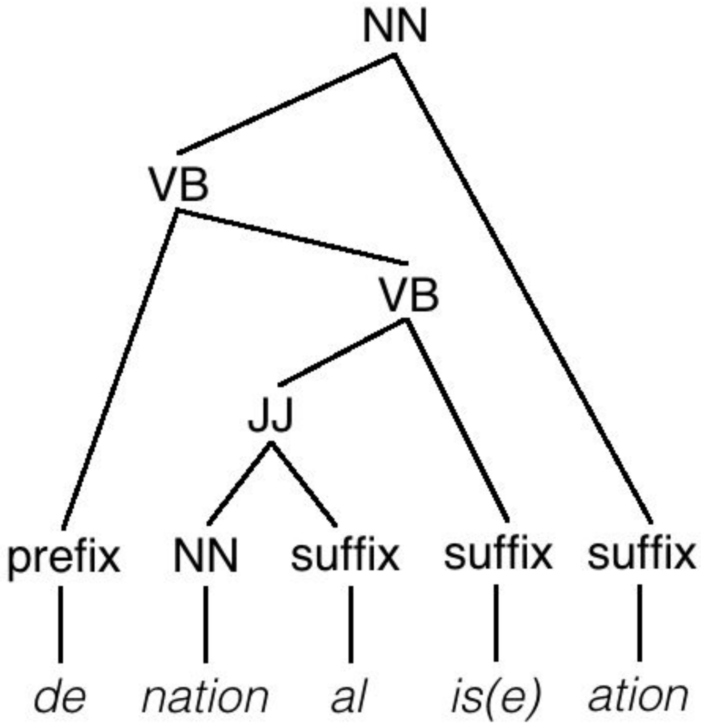
- *lawlessness*
- *ghostlike*
- *mistreatment*
- *bittersweet*
- *satisfactory*
- *mouth-watering*

2. Word Length

Some long words are actually simple:

- *law + less + ness*
- *ghost + like*
- *mis + treat + ment*
- *bitter + sweet*
- *satisf(y) + act + ory*
- *mouth + - + watering*

Derivational Morphology



3. Subword Features

Complex words have rare letter combinations:

- *abhorrence*
  - 5-grams: ^*abho, abhor, bhorr, ..., rence, ence$*
  - 4-grams: ^*abh, abho, bhor, horr, ..., ence, nce$*
  - 3-grams: ^*ab, abh, bho, hor, orr, ..., enc, nce, ce$*

3. Subword Features

Compare:

- *abhorrence*
  - 4-grams: *^abh, abho, bhor, horr, ..., ence, nce$*
- *anger*
  - 4-grams: *^ang, ange, nger, ger$*

4. Phonetic Features

Complex words have higher consonant-vowel ratio:

- *procrastinate*
- *flabbergasted*

- *neighbourhood*
- *information*

4. Phonetic Features

Complex words have higher consonant-vowel ratio:

- */prəˈkræstəneɪt/ - 8 consonants vs. 5 vowels*
- */ˈflæbəgɑːstɪd/ - 7 consonants vs. 4 vowels*

- */ˈneɪbəhʊd/ - 4 consonants vs. 4 vowels*
- */ˌɪnfəˈmeɪʃən/ - 5 consonants vs. 5 vowels*

4. Phonetic Features

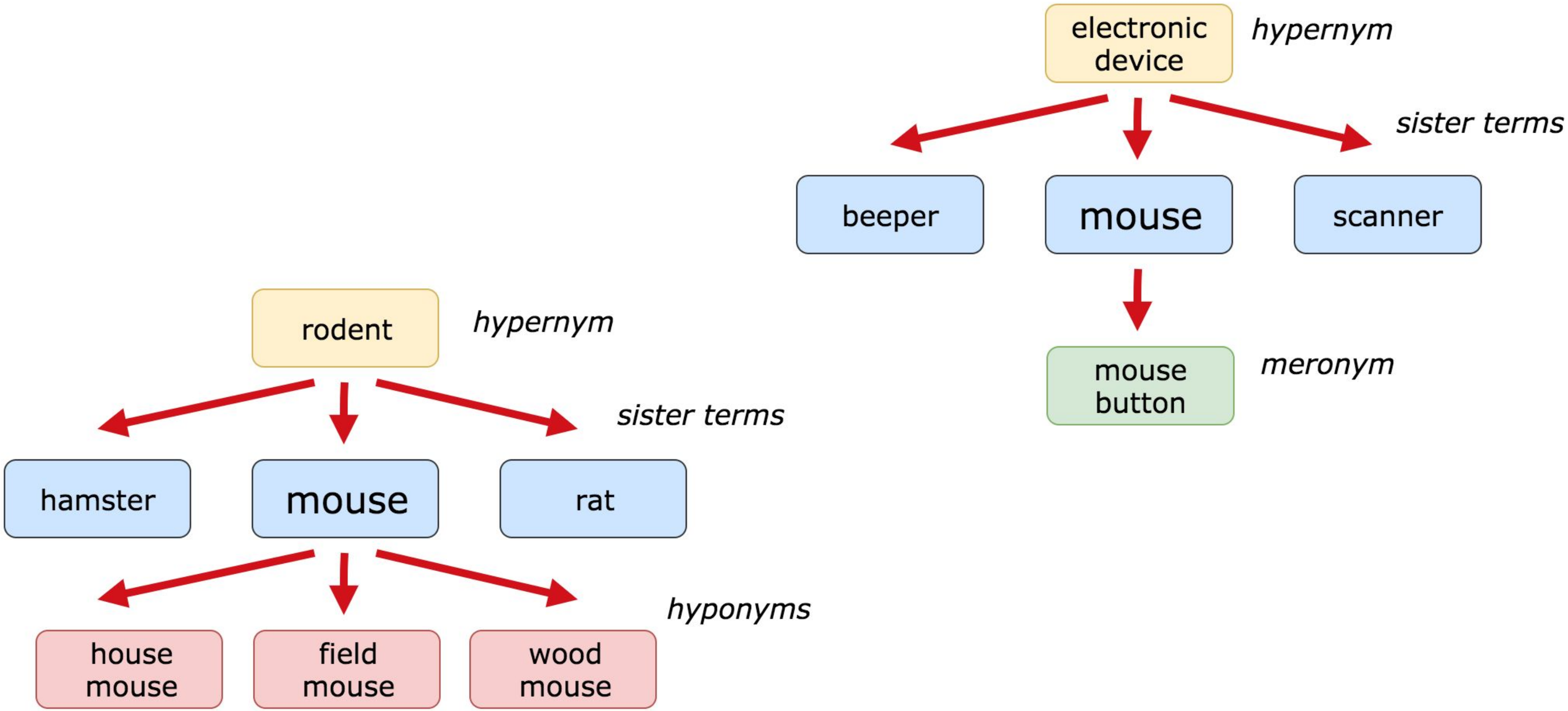
- number of vowels
- number of consonants
- ratio of consonants vs. vowels
- number of repeating sounds
- number of syllables

5. Semantic Features

5. Semantic Features

Word	Number of senses in WordNet
report	7 n + 6 v
mouse	4 n + 2 v
elucidate	2 v
moribund	2 a
abhorrence	1 n

5. Semantic Features



5. Semantic Features

- number of senses
- number of hypernyms
- number of hyponyms
- number of holonyms
- number of meronyms

6. Psycholinguistic Features

- concreteness
- imageability
- familiarity
- age of acquisition

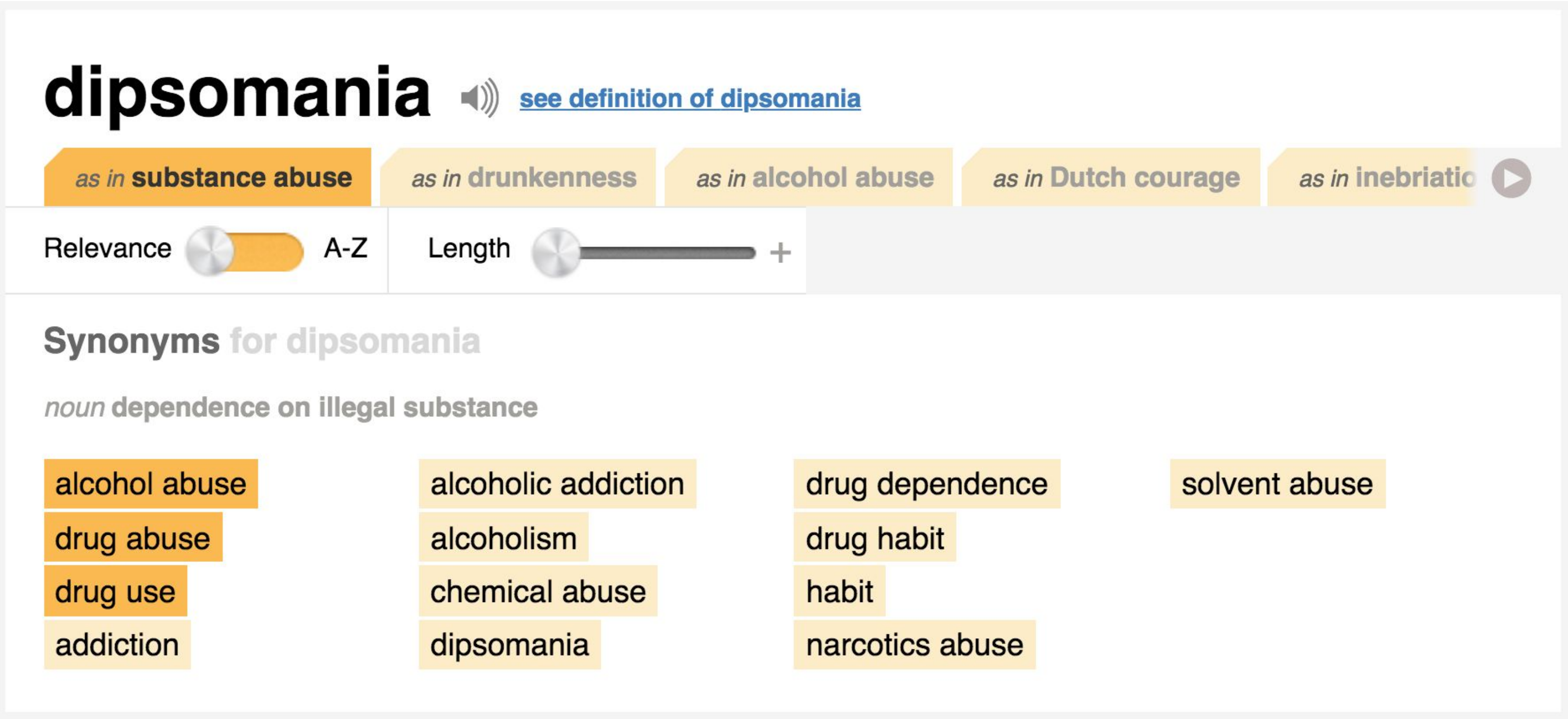
E.g., see *MRC Psycholinguistic Database*.

4. Complex word simplification

Approach

- lemmatize the word
- detect part of speech
- *do word sense disambiguation*
- extract synonyms from a thesaurus
- put synonyms in place of the original word
- rank (how?)

Candidate replacements



Taken from https://www.thesaurus.com/

Candidate replacements

- *His dipsomania led to the loss of his loved one.*
- *His habit led to the loss of his loved one.*
- *His alcoholism led to the loss of his loved one.*
- *His alcohol abuse led to the loss of his loved one.*
- *His insobriety led to the loss of his loved one.*
- *His inebriacy led to the loss of his loved one.*
- ...

Questions

- Is it simpler?
  - *use our CWI module*
- Is it not too simple?
  - *check polysemy*
  - *check frequency*
- Is it grammatically correct?
  - *it depends...*

Is it grammatically correct?

- correct verb form
  - *affront* => *insult*
  - *affronts* => *insults*
  - *affronted* => *insulted*
  - ...

Is it grammatically correct?

- correct noun number
  - *revelries => celebrations, festivities*
- article change
  - *a destitute area => an impoverished area*
- degrees of comparison
  - *more destitute => poorer*
  - *brawnier => more muscular*

Is it grammatically correct?

- governing
  - *She knew about the plan and colluded with him.*
  - *She knew about the plan and conspired with him.*

  - *She knew about the plan and colluded in it.*
  - ** She knew about the plan and conspired in it.*

How to rank the suggestions?

Use a language model:

- orig = *"The patient was moribund."*
- repl_1 = *"The patient was dying."*
- repl_2 = *"The patient was fading."*
- …
- repl_n = *"The patient was declining."*

Which replacement is the most fitting?

Language modelling

- Statistical language modelling
- Neural language modelling

Language modelling

- Statistical language modelling
- Neural language modelling

Chain rule

P(“*<S> The patient was fading . </S>*”) =

Chain rule

P(“<S> *The patient was fading . </S>”) =*

*P(“The”|”<S>”) **

Chain rule

P(*“<S> The patient was fading . </S>”) =*

*P(“The”|”<S>”) **

*P(“patient”|”<S> The”) **

Chain rule

P(*“<S> The patient was* *fading . </S>”) =*

*P(“The”|”<S>”) **

*P(“patient”|”<S> The”) **

*P(“was”|”<S> The patient”) **

Chain rule

P(“<S> The patient was fading . </S>”) =

P(“The”|”<S>”) *

P(“patient”|”<S> The”) *

P(“was”|”<S> The patient”) *

P(“fading”|“<S> The patient was”) *

P(“.”|“<S> The patient was fading”)

Markov assumption

The future is independent of the past given the present.

Chain rule

P(“<S> The patient was fading . </S>”) =

P(“The”|”<S>”) *

P(“patient”|”<S> The”) *

P(“was”|”<S> The patient”) *

P(“fading”|“<S> The patient was”) *

P(“.”|“<S> The patient was fading”)

Markov assumption

P(“<S> *The patient was fading . </S>*”) =

*P(“The”|”<S>”) **

*P(“patient”|”The”) **

*P(“was”|”patient”) **

*P(“fading”|“was”) **

P(“.”|“fading”)

Markov assumption

P("<S> *The patient was fading . </S>*") =

*P("The" | "<S>") **

*P("patient" | "The") **

*P("was" | "patient") **

*P("fading" | "was") **

P("." | "fading")

Markov assumption

P("<S> The patient was fading . </S>") =

*P("The" | "<S>") **

P("patient" | "The") = C("The patient") / C("The")

*P("was" | "patient") **

*P("fading" | "was") **

P("." | "fading")

What if we never saw “fading”?

P(“*<S> The patient was fading . </S>*”) = *0 ?*

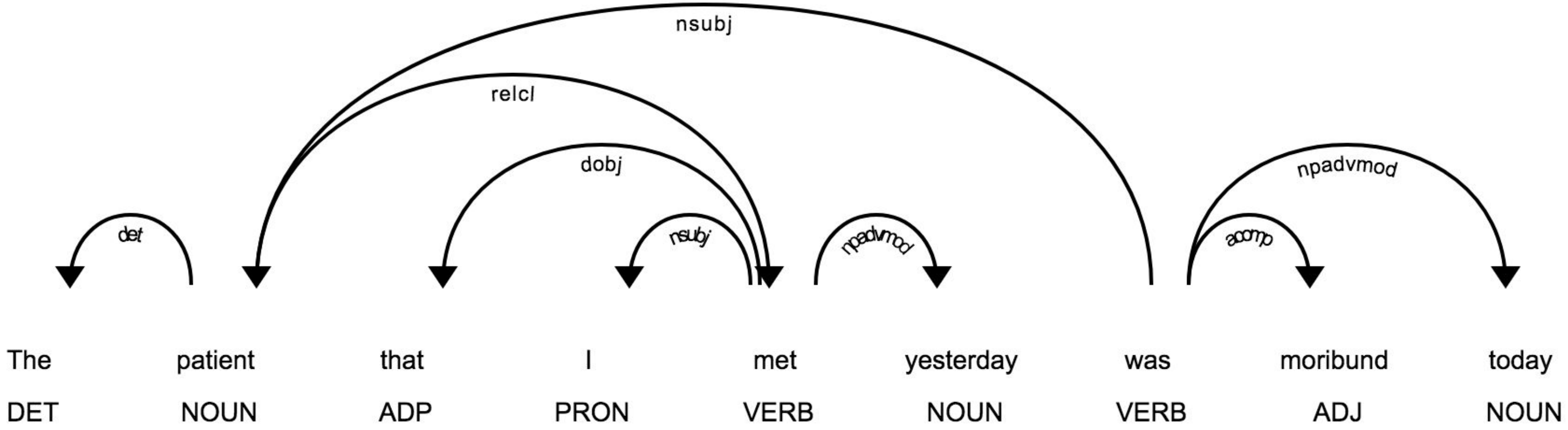
Smoothing techniques

- Add-1 smoothing
- Add-k smoothing
- Backoff
- Interpolation
- Kneser-Ney smoothing
- …

Statistical LM challenges

- Does not generalize
  - *C("red car") = 2 390*
  - *C("blue car") = 1 113*
  - *C("purple car") = 0*
- Does not capture long-range dependencies
  - *The patient that I met yesterday was moribund today.*
- Scaling to larger ngrams is very expensive
- Needs intricate smoothing techniques

LM with syntactic ngrams

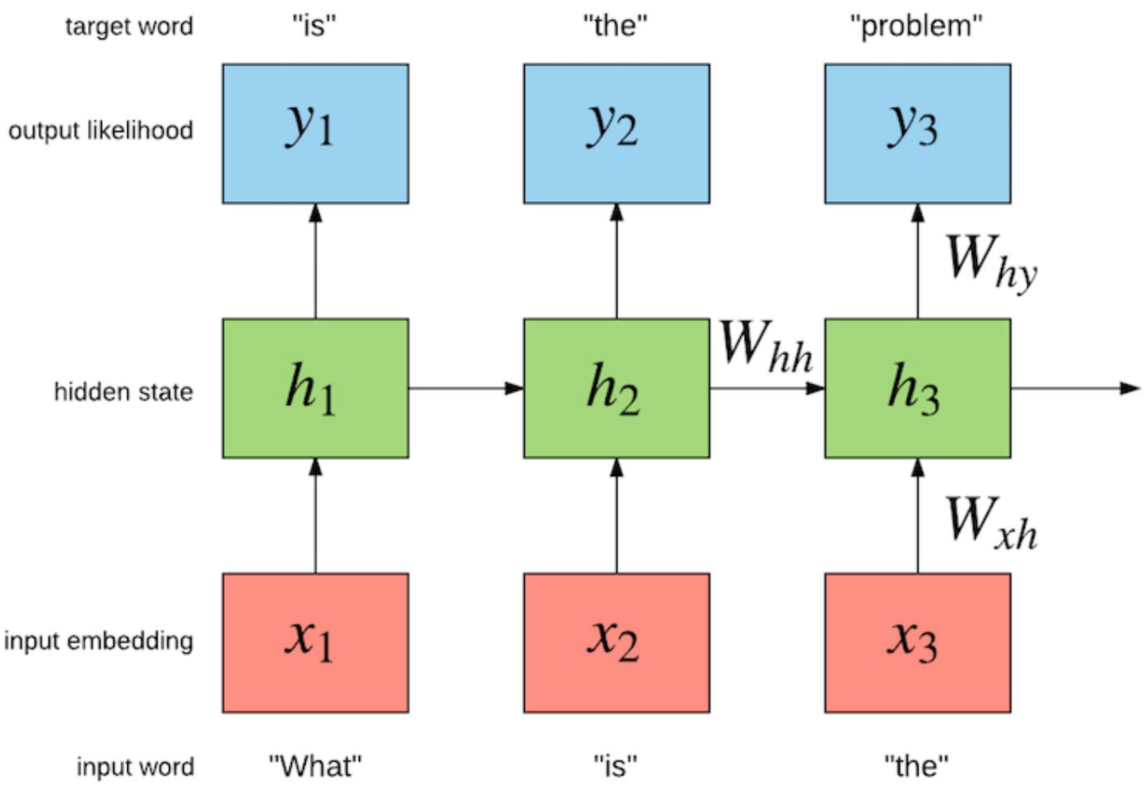


Built by https://explosion.ai/demos/displacy

Language modelling

- Statistical language modelling
- Neural language modelling

Neural language modelling



Taken from http://torch.ch/blog/2016/07/25/nce.html

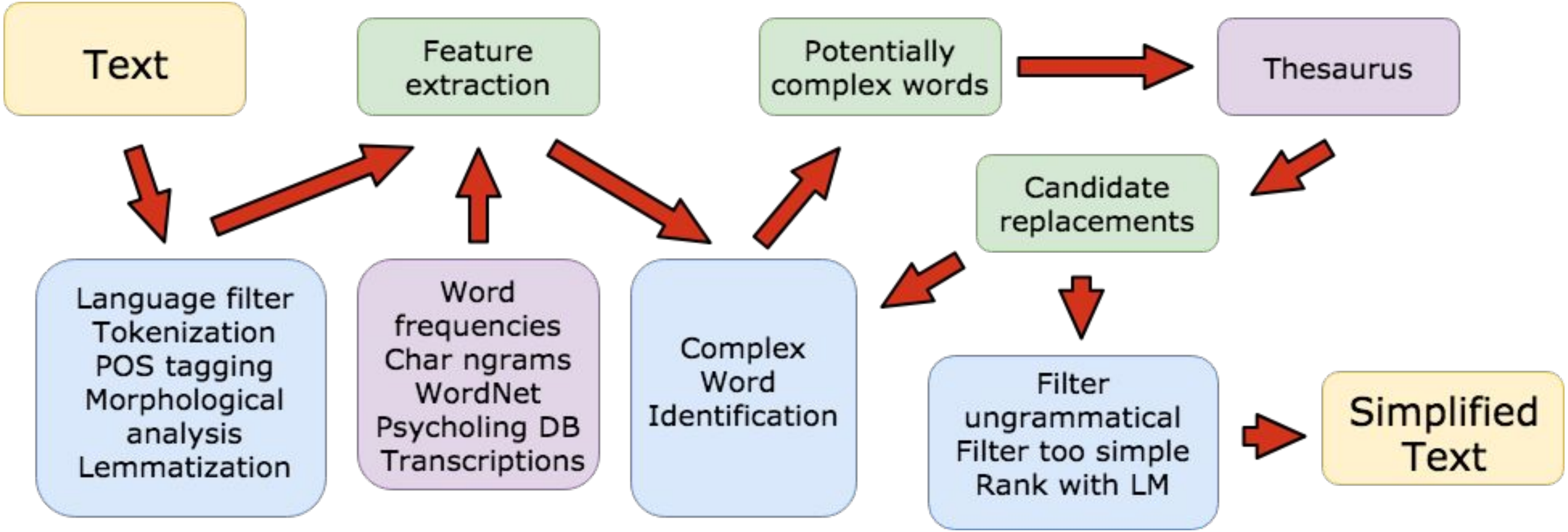
Neural LM challenges

- Takes a long time to train
- Much more expensive
- May generalize too much
  - *brown horse, white horse, green horse O_o*
- Not too much improvement

Candidate replacements

- *His dipsomania led to the loss of his loved one.*
- *His alcoholism led to the loss of his loved one.*

Final pipeline



Conclusion

Three things I’d like to highlight

1. Linguistic knowledge gives you power.
2. Researchers are not the final consumers of NLP applications.
3. Diving into the problem gives better results that not diving into the problem.

Q: When your ML doesn't outperform, what do you do?

Your turn: select an answer.

1. Sigh.
2. Gather more data. Maybe your model needs more data to find proper weights.
3. Simplify your model. Maybe its easier just to do more hyperparameter searching.
4. Read your arXiv feed.

Taken from http://bit.ly/kan-coling18

Q: When your ML doesn't outperform, what do you do?

Your turn: select an answer.

1. Sigh.
2. Gather more data. Maybe your model needs more data to find proper weights.
3. Simplify your model. Maybe its easier just to do more hyperparameter searching.
4. Read your arXiv feed.

5. Study your problem more. Let your text speak.

Taken from http://bit.ly/kan-coling18

Thank you !

Any questions ?

mariana.romanyshyn@grammarly.com

Useful links

- Libraries
  - https://spacy.io/
  - https://stanfordnlp.github.io/CoreNLP/
- Resources
  - https://wordnet.princeton.edu/
  - http://thesaurus.com/
  - https://en.wiktionary.org/
  - MRC Psycholinguistic Database

Useful links

- Ngrams:
  - https://books.google.com/ngrams
  - https://www.ngrams.info/
- Language models:
  - https://github.com/kpu/kenlm
  - https://github.com/pytorch/examples/tree/master/word_language_model
  - https://github.com/salesforce/awd-lstm-lm