Natural Language Processing (NLP) with R

Thursday 27th June, 2019

Typical NLP tasks

- Tokenization
- Sentence splitting
- Part-of-speech (POS) tagging
- Lemmatization
- Named entity recognition
- Parsing
  - Constituency Parsing
  - Dependency Parsing
- Sentiment analysis
- Coreference Resolution
- ...

Motivation

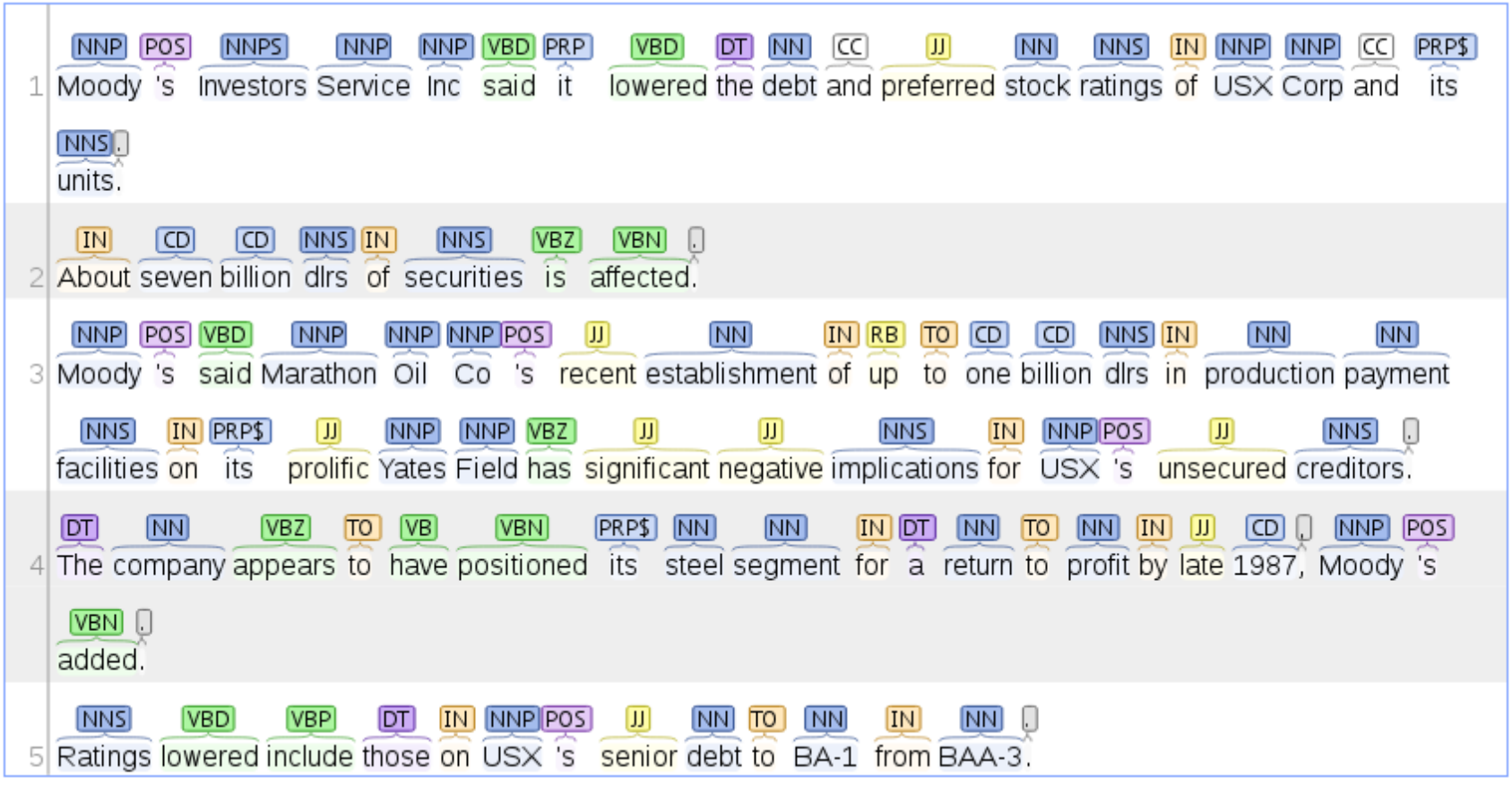


Figure: Part-of-speech (POS) tags for a text from the Reuters21578 corpus.

Penn Treebank part-of-speech tags (including punctuation)

Tag	Description	Example	Tag	Description	Example
CC	coordin. conjunction	*and, but, or*	SYM	symbol	+,%, &
CD	cardinal number	*one, two*	TO	"to"	*to*
DT	determiner	*a, the*	UH	interjection	*ah, oops*
EX	existential 'there'	*there*	VB	verb base form	*eat*
FW	foreign word	*mea culpa*	VBD	verb past tense	*ate*
IN	preposition/sub-conj	*of, in, by*	VBG	verb gerund	*eating*
JJ	adjective	*yellow*	VBN	verb past participle	*eaten*
JJR	adj., comparative	*bigger*	VBP	verb non-3sg pres	*eat*
JJS	adj., superlative	*wildest*	VBZ	verb 3sg pres	*eats*
LS	list item marker	*1, 2, One*	WDT	wh-determiner	*which, that*
MD	modal	*can, should*	WP	wh-pronoun	*what, who*
NN	noun, sing. or mass	*llama*	WP$	possessive wh-	*whose*
NNS	noun, plural	*llamas*	WRB	wh-adverb	*how, where*
NNP	proper noun, sing.	*IBM*	$	dollar sign	*$*
NNPS	proper noun, plural	*Carolinas*	#	pound sign	*#*
PDT	predeterminer	*all, both*	"	left quote	' or "
POS	possessive ending	*'s*	"	right quote	' or "
PRP	personal pronoun	*I, you, he*	(	left parenthesis	[, (, {, <
PRP$	possessive pronoun	*your, one's*	)	right parenthesis	],), }, >
RB	adverb	*quickly, never*	,	comma	,
RBR	adverb, comparative	*faster*	.	sentence-final punc	. ! ?
RBS	adverb, superlative	*fastest*	:	mid-sentence punc	: ; ... – -
RP	particle	*up, off*			

Figure 10.1 Penn Treebank part-of-speech tags (including punctuation).

Figure: Source: `https://web.stanford.edu/~jurafsky/slp3/8.pdf`

Motivation

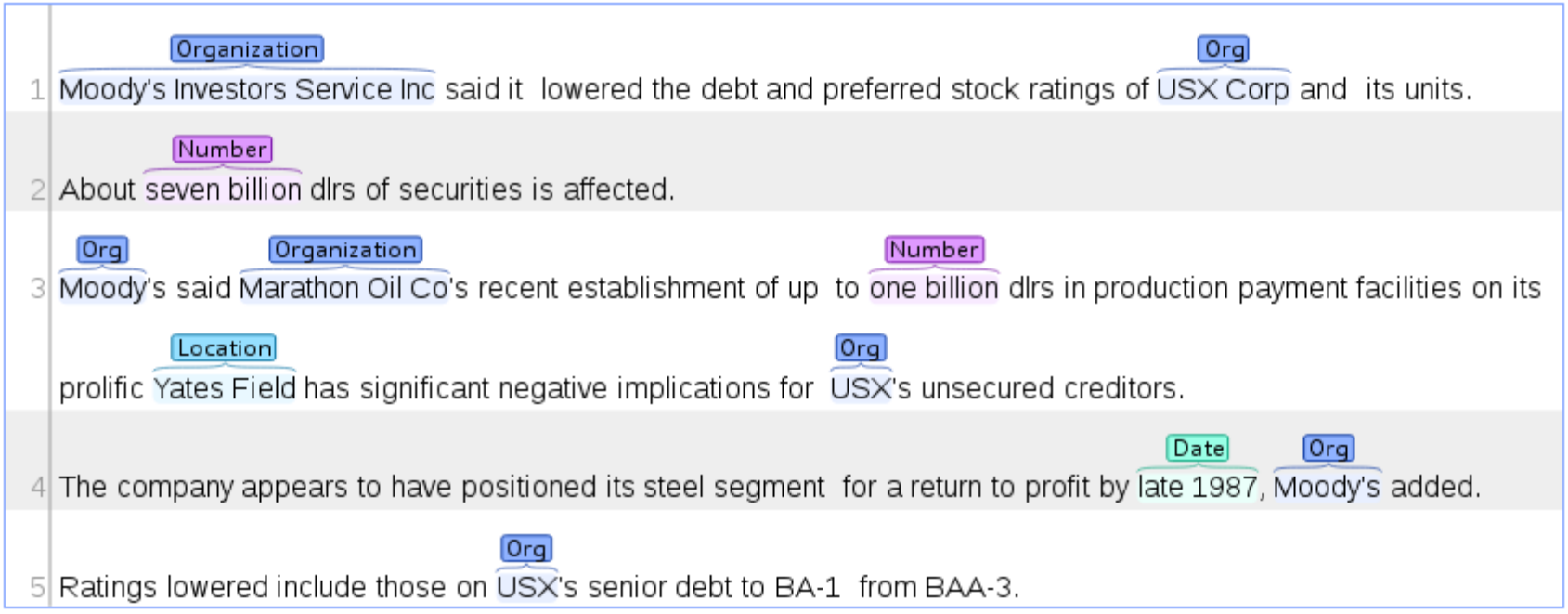


Figure: Named entity annotation for a text from the Reuters21578 corpus.

Motivation

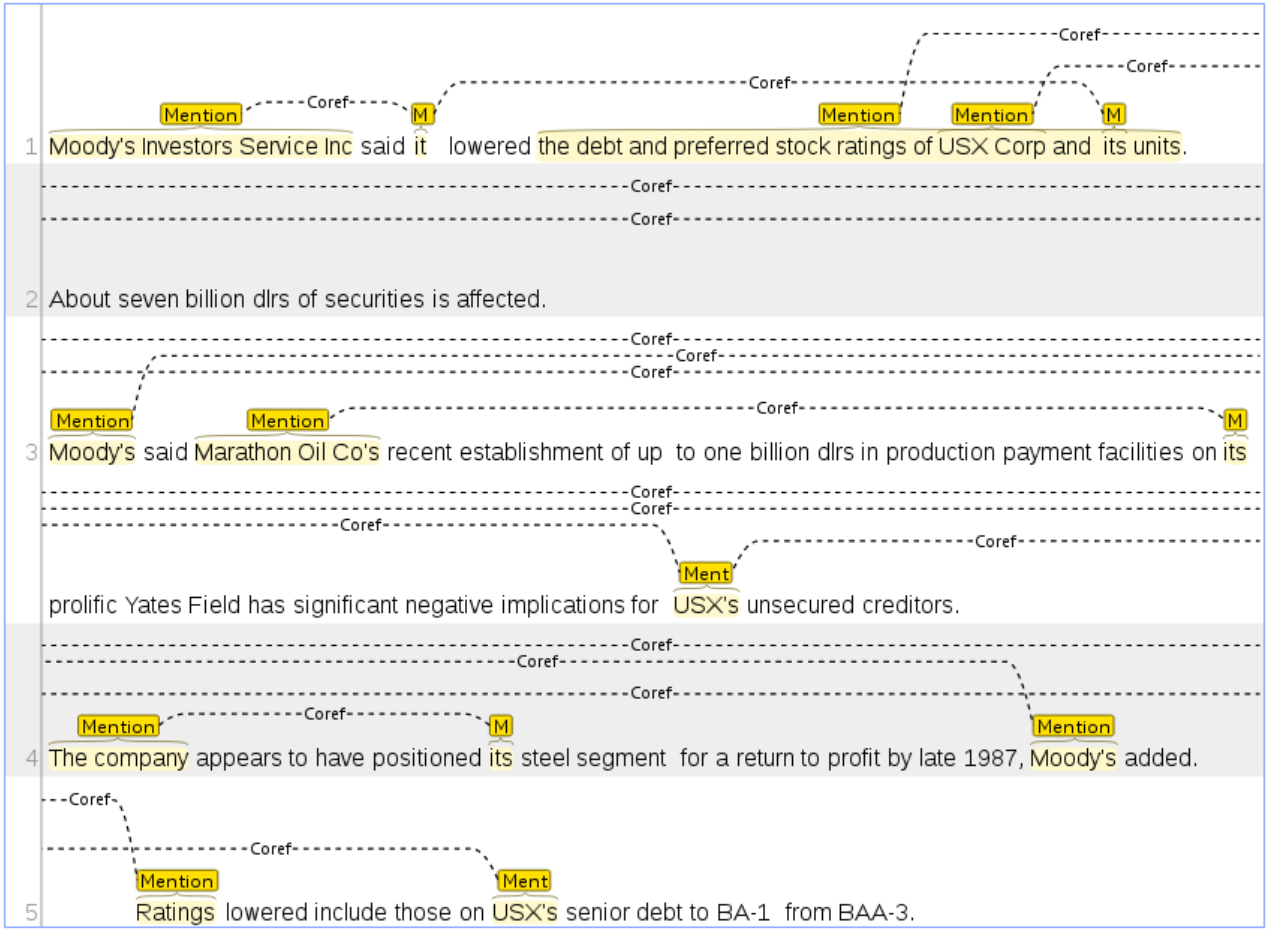


Figure: Coreference annotations for a text from the Reuters21578 corpus.

NLP tools available in R

Software	Prog. lang.	Languages	R-wrapper
Stanford CoreNLP	Java	ar, de, en, es, fr, zh	**StanfordCoreNL** **coreNLP**
OpenNLP	Java	da, de, en, es, it, nl, pt, sv	**OpenNLP**
spaCy	Python	de, en, es, fr, it, nl, pt	**spacyr**
UDPipe	C++	> 50	**udpipe**
Google API	REST-API	de, en, es, fr, it, ja, ko, pt, zh	**googlenlp**

Table: NLP resources in R

R-NLP infrastructures

cleanNLP (Arnold, 2017)

- Imports + Suggests: **dplyr**, **Matrix**, **stringi**, **udpipe**, **reticulate**, **rJava**, **RCurl**, ...
- SystemRequirements: Java, Python

NLP (Hornik, 2018a)

- Imports + Suggests: **utils**
- SystemRequirements:

	cleanNLP	**NLP**
OpenNLP		✓
spaCy	✓	(✓)
Stanford CoreNLP	✓	✓
UDPipe	✓	(✓)

NLP with the **StanfordCoreNLP** package

Installation

```
install.packages("NLP")
install.packages("rJava")
install.datacube <- function(pkg) install.packages(pkg,
  repos = "http://datacube.wu.ac.at/", type = "source")

install.datacube("StanfordCoreNLP")
install.datacube("StanfordCoreNLPjars")    ## en - models
install.datacube("StanfordCoreNLPjars.de") ## de - models
```

Load

```
options(java.parameters = "-Xmx4g")
library("NLP")
library("StanfordCoreNLP")
```

NLP with the **StanfordCoreNLP** package

The following example text contains the first four sentences from an article from telegraph.co.uk.

```
txt <- "I know words. I have the best words. Donald Trump
  said one day in his superlative way. Now those words by
  the new US president have been pulled together as a
  collection of poetry in Norway."
```

Annotate

```
pline <- StanfordCoreNLP_Pipeline(
  annotators = c("tokenize", "ssplit", "pos", "lemma",
    "ner", "parse", "sentiment", "dcoref"))

a <- AnnotatedPlainTextDocument(txt, annotate(txt, pline))
```

Tokenization & Sentence splitting

Word tokens

```
words(a)[1:10]

##  [1] "I"     "know"  "words" "."     "I"     "have"
##  [7] "the"   "best"  "words" "."
```

Sentences

```
sents(a)[1:2]

## [[1]]
## [1] "I"     "know"  "words" "."
##
## [[2]]
## [1] "I"     "have"  "the"   "best"  "words" "."
```

Part-of-speech (POS) tagging

Part-of-speech tagging is the task of assigning the correct part of speech tag (noun, verb, etc.) to words.

Part-of-speech (POS) tagging

Part-of-speech tagging is the task of assigning the correct part of speech tag (noun, verb, etc.) to words.

- accuracy token level is around 97%
- accuracy sentence level is around 57%

Part-of-speech (POS) tagging

Part-of-speech tagging is the task of assigning the correct part of speech tag (noun, verb, etc.) to words.

Part of speech tags

```
tagged_words(a)[1:10]

## I/PRP
## know/VBP
## words/NNS
## ./.
## I/PRP
## have/VBP
## the/DT
## best/JJS
## words/NNS
## ./.
```

Lemmatization

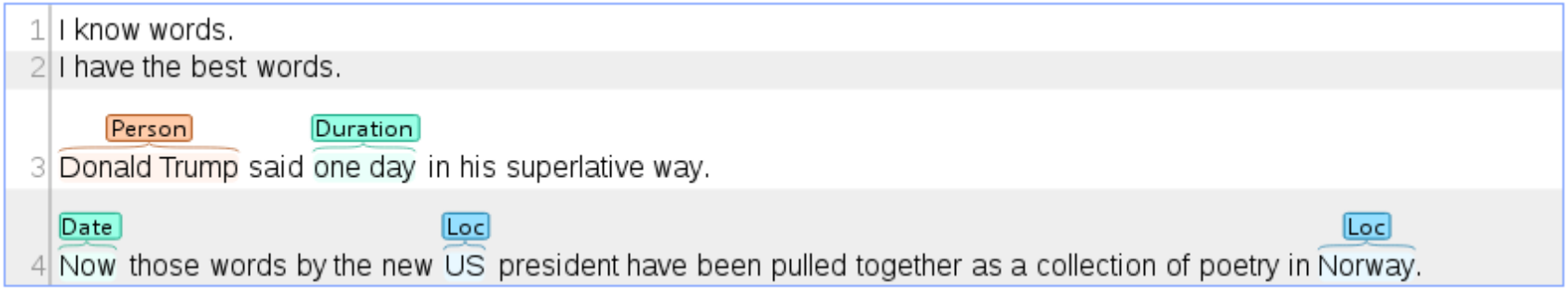
Lemmas

```
lem <- features(a, "word")$lemma
cbind(words = words(a), lemmas = lem)[12:20,]

##       words          lemmas
##  [1,] "Trump"        "Trump"
##  [2,] "said"         "say"
##  [3,] "one"          "one"
##  [4,] "day"          "day"
##  [5,] "in"           "in"
##  [6,] "his"          "he"
##  [7,] "superlative" "superlative"
##  [8,] "way"          "way"
##  [9,] "."            "."
```

Named entity recognition

- **proper name:** PERSON, LOCATION, ORGANIZATION, MISC
- **numerical:** MONEY, NUMBER, ORDINAL, PERCENT
- **temporal:** DATE, TIME, DURATION



Named entity recognition

Named entities

```
ner <- features(a, "word")$NER
cbind(id = seq_along(ner), words = words(a),
      ner = ner)[ner != "O",]

##      id   words       ner
## [1,] "11" "Donald"    "PERSON"
## [2,] "12" "Trump"     "PERSON"
## [3,] "14" "one"       "DURATION"
## [4,] "15" "day"       "DURATION"
## [5,] "21" "Now"       "DATE"
## [6,] "27" "US"        "COUNTRY"
## [7,] "28" "president" "TITLE"
## [8,] "39" "Norway"    "COUNTRY"
```

Syntactic parsing (phrase structure grammar)

Parse trees (Syntax trees) are used to analyze (represent) the structure of a sentence.

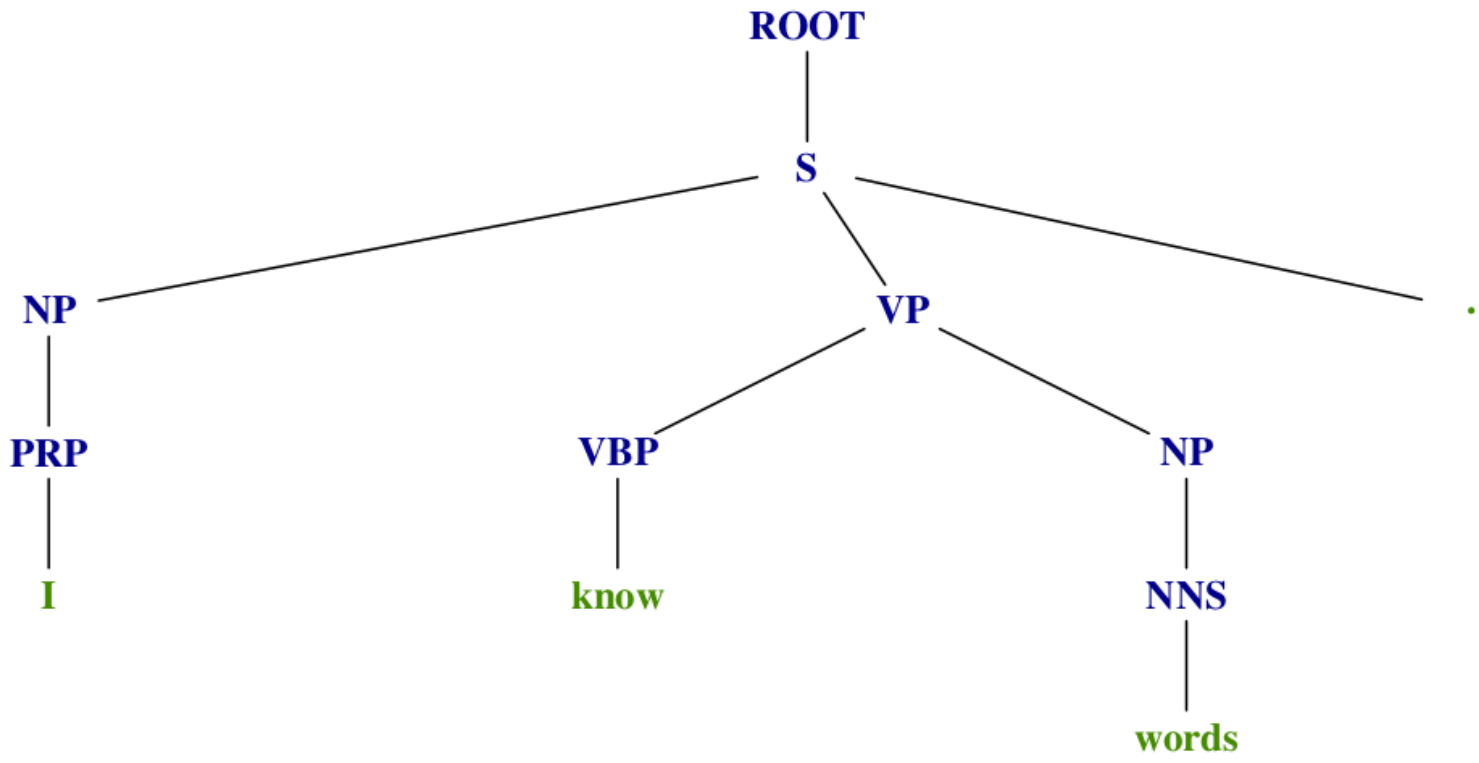


Figure: I know words.

Syntactic parsing (phrase structure grammar)

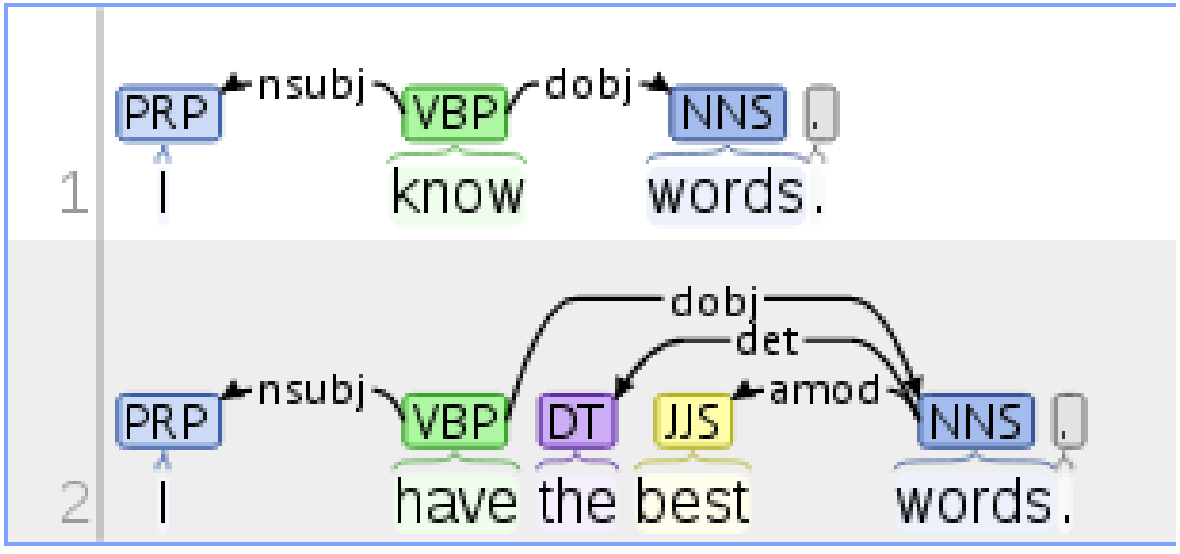
Parse

```
parsed_sents(a)[[1L]]

## (ROOT
##   (S
##     (NP (PRP I))
##     (VP (VBP know) (NP (NNS words)))
##     (. .)))
```

Dependency Parsing

- Dependency structure shows which words depend on (modify or are arguments of) which other words.
- Is used to analyze the relation between a word and its dependents.



Dependency Parsing

Basic dependencies

```
features(a, "sentence")[["basic-dependencies"]][[2]]

## root(ROOT-0, have-2)
## nsubj(have-2, I-1)
## det(words-5, the-3)
## amod(words-5, best-4)
## dobj(have-2, words-5)
## punct(have-2, .-6)
```

Sentiment analysis

Sentiment

```
features(a, "sentence")[c("sentiment", "sentimentValue")]

##   sentiment sentimentValue
## 1   Neutral              2
## 2  Positive              3
## 3   Neutral              2
## 4   Neutral              2
```

Coreference resolution

Coreferences

```
features(a, "document")$coreferences[[1L]]

## [[1]]
##   representative sentence start end head text
## 1           TRUE        4     7   7    7   US
## 2          FALSE        1     1   1    1    I
## 3          FALSE        2     1   1    1    I
##
## [[2]]
##   representative sentence start end head          text
## 1           TRUE        3     1   2    2 Donald Trump
## 2          FALSE        3     7   7    7           his
```

NLP as data preparation step

- Sentence splitting is used to estimate topic models on a sentence level.
- POS-tags are used to identify words to be removed during the data preparation of classification tasks (e.g. topic models).
- Lemmatization and the identification of compounds are used as a data preparation step in classification tasks.
- Named entity recognition is used to extract additional features from text.
- ...

Taylor Arnold. A tidy data model for natural language processing using **cleanNLP**. *The R Journal*, 9(2):1–20, 2017. URL https://journal.r-project.org/archive/2017/RJ-2017-035/index.html.

Kurt Hornik. **NLP**: *Natural Language Processing Infrastructure*, 2018a. R package version 0.1-11.5.

Kurt Hornik. **StanfordCoreNLP**: *Stanford CoreNLP Annotation.*, 2018b. URL https://datacube.wu.ac.at. R package version 0.1-4.2.

Daniel Jurafsky and James H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. 3rd ed. draft edition, 2017. URL https://web.stanford.edu/~jurafsky/slp3/ed3book.pdf.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60, 2014. URL http://www.aclweb.org/anthology/P/P14/P14-5010.

Pontus Stenetorp, Goran Topić, Sampo Pyysalo, Tomoko Ohta, Jin-Dong Kim, and Jun'ichi Tsujii. Bionlp shared task 2011: Supporting resources. In *Proceedings of BioNLP Shared Task 2011 Workshop*, pages 112–120, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W11-1816.