

## Learner Corpus Research Summer School

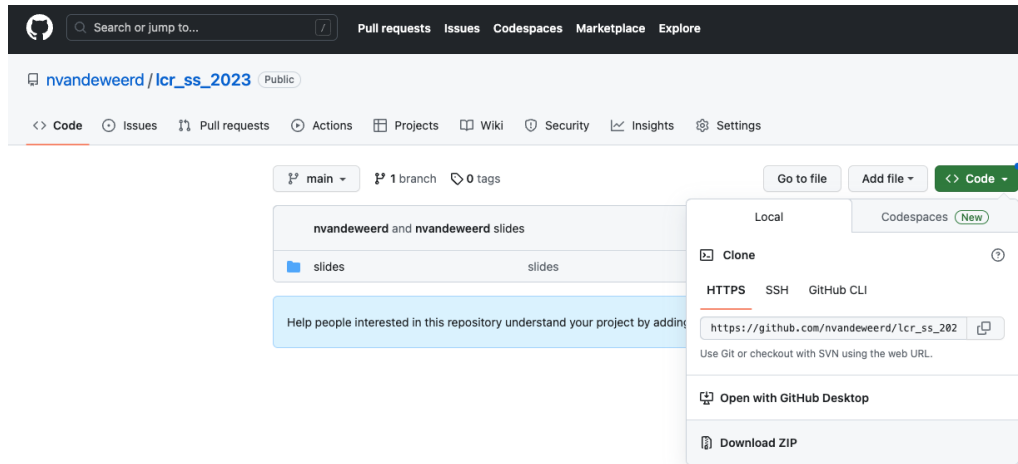
Automatic annotation of learner corpus data

Dr. Nathan Vandeweerd

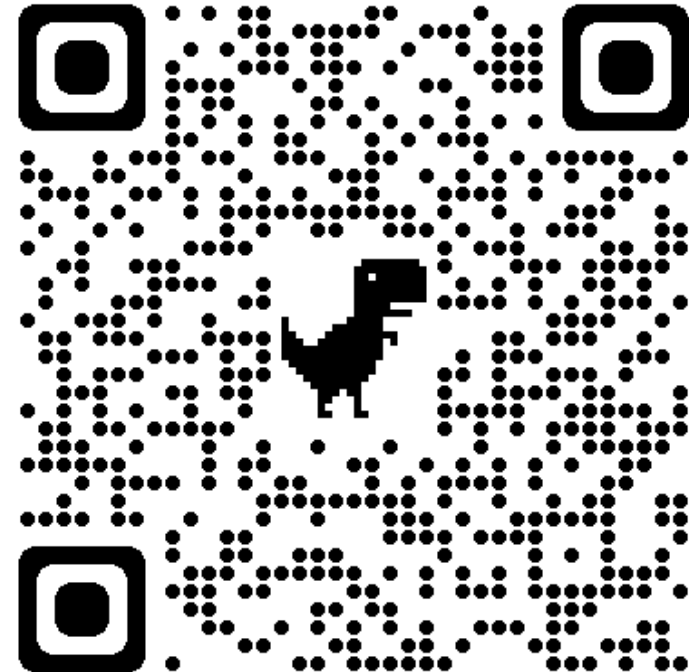
Louvain-la-Neuve, Belgium

July 4th, 2023

# All materials and sides available on GitHub



1. Click on **< > Code**.
2. Download ZIP to download all files.



# Introduction

# About me

🏛️ Radboud University Nijmegen

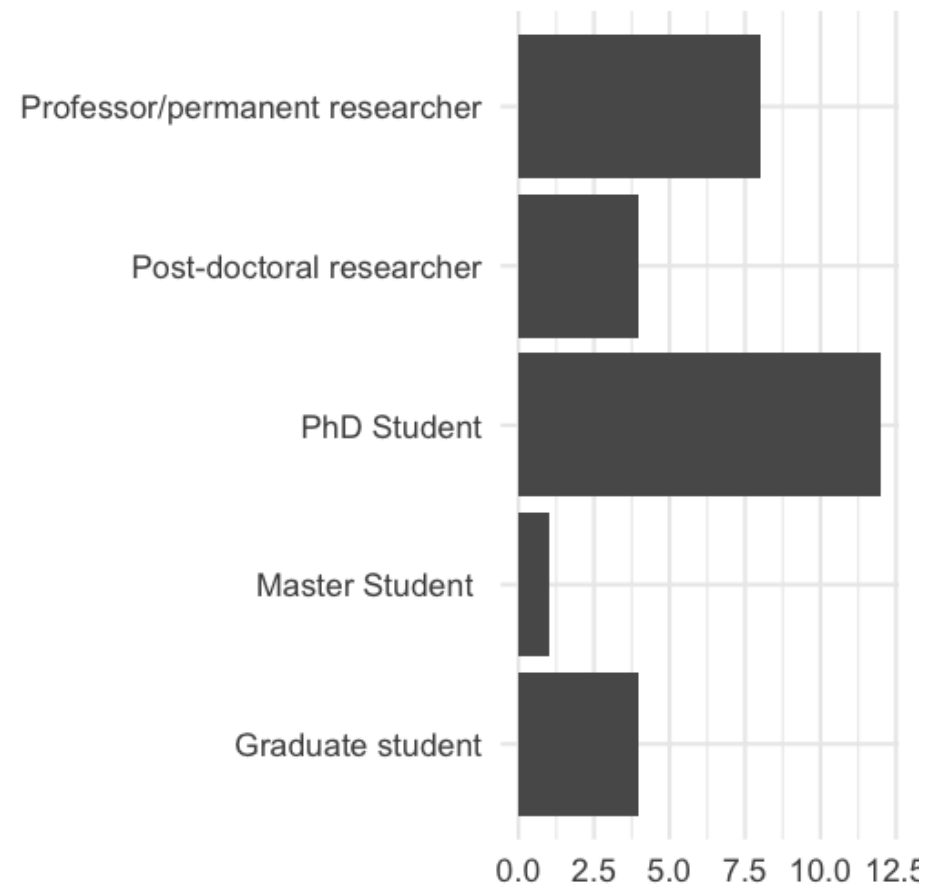
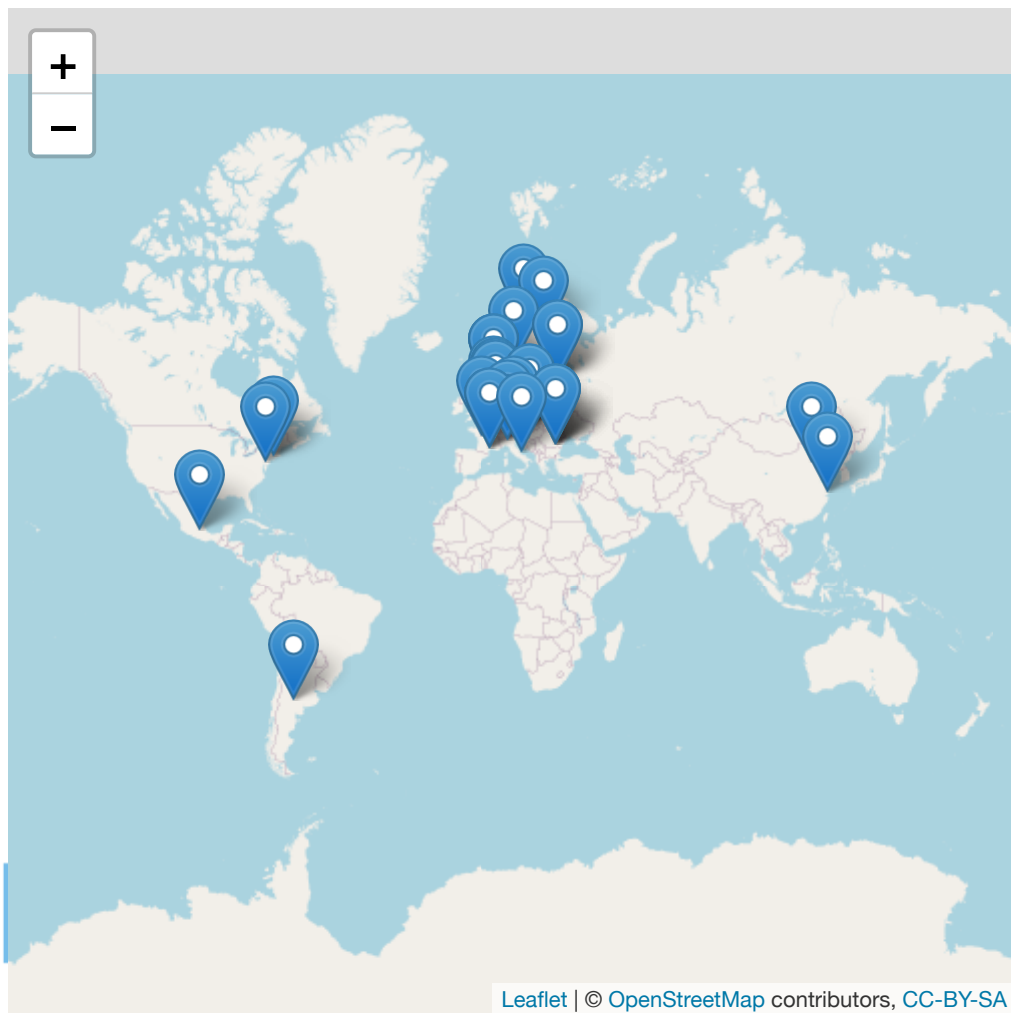
🏢 Assistant professor in *Language and Communication*

🔍 Research interests:

- Phraseological complexity in L2 French
- Accuracy of automatic transcription software for L2 data
- Language development during study abroad
- Crowdsourcing language assessment project



# About you



# Your goals

"the correlation between corpus building and its exploitation"

"most appropriate scientific approaches that one may use in analysing data from a learner corpus"

"I would like to know which new (or less new) software might be used to analyse the corpus. "

"how to build my own machine-learning scheme when using topic modelling?"

"Secondly, I wish to get some theoretical knowledge & practical skills on programming"

# What types of automatic annotation exist?

1. Part of Speech (POS) tagging



# What types of automatic annotation exist?

## 1. Part of Speech (POS) tagging

(1) We\_PPIS2 find\_VV0 that\_CST in\_II fact\_NN1 these\_DD2 people\_NN  
are\_VBR the\_AT most\_RGT exposed\_JJ to\_II media\_NN not\_XX to\_TO  
mension\_VVI the\_AT fact\_NN1 that\_CST there\_EX is\_VBZ forever\_RT  
AIDS\_NP1 awareness\_NN1 campaigns\_NN2 launged\_VVN through\_RP  
out\_RP the\_AT county\_NN1.\_

(ICLE-TS-NOUN-0005.1)

(van Rooy, 2015: 80)

Q Your research:

- Spanish articles
- morpho-syntactic and syntactic labels
- morphology
- semantic, syntactic and discourse features
- verb valency patterns
- grammatical complexity
- grammatical case
- cohesion and cohesive devices
- stance features
- verb aspect
- adverbs of degree and negation
- verb phrase ellipsis
- colour terms
- meta-discourse markers
- formulaic language



# What types of automatic annotation exist?

## 1. Part of Speech (POS) tagging

(1) We\_PPIS2 find\_VV0 that\_CST in\_II fact\_NN1 these\_DD2 people\_NN  
are\_VBR the\_AT most\_RGT exposed\_JJ to\_II media\_NN not\_XX to\_TO  
mension\_VVI the\_AT fact\_NN1 that\_CST there\_EX is\_VBZ forever\_RT  
AIDS\_NP1 awareness\_NN1 campaigns\_NN2 launged\_VVN through\_RP  
out\_RP the\_AT county\_NN1.\_

(ICLE-TS-NOUN-0005.1)

(van Rooy, 2015: 80)

Q Your research:

- Spanish **articles**
- **morpho-syntactic and syntactic labels**
- **morphology**
- semantic, syntactic and **discourse features**
- **verb** valency patterns
- grammatical complexity
- grammatical **case**
- cohesion and **cohesive devices**
- stance features
- **verb aspect**
- **adverbs** of degree and negation
- **verb** phrase ellipsis
- colour terms
- meta-discourse markers
- formulaic language

# What types of automatic annotation exist?

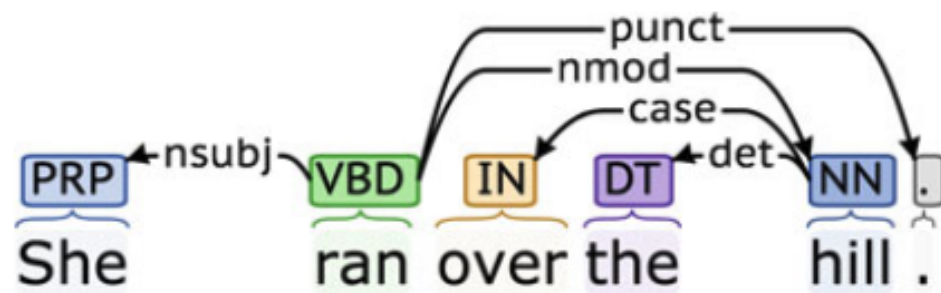
1. Part of Speech (POS) tagging
2. Syntactic parsing

## Q Your research:

- Spanish **articles**
- **morpho-syntactic and syntactic labels**
- **morphology**
- semantic, syntactic and **discourse features**
- **verb** valency patterns
- grammatical complexity
- grammatical **case**
- cohesion and **cohesive devices**
- stance features
- **verb aspect**
- **adverbs** of degree and negation
- **verb** phrase ellipsis
- colour terms
- meta-discourse markers
- formulaic language

# What types of automatic annotation exist?

1. Part of Speech (POS) tagging
2. Syntactic parsing



(Newman and Cox, 2021: 32)

## Q Your research:

- Spanish **articles**
- **morpho-syntactic** and **syntactic** labels
- **morphology**
- semantic, **syntactic** and **discourse** features
- **verb valency patterns**
- **grammatical complexity**
- grammatical **case**
- cohesion and **cohesive devices**
- stance features
- **verb aspect**
- **adverbs** of degree and negation
- **verb phrase** ellipsis
- colour terms
- meta-discourse markers
- **formulaic language**

# What types of automatic annotation exist?

1. Part of Speech (POS) tagging
2. Syntactic parsing
3. Semantic annotation

## Q Your research:

- Spanish **articles**
- **morpho-syntactic** and **syntactic** labels
- **morphology**
- semantic, **syntactic** and **discourse** features
- **verb valency** patterns
- **grammatical complexity**
- grammatical **case**
- cohesion and **cohesive devices**
- stance features
- **verb aspect**
- **adverbs** of degree and negation
- **verb phrase** ellipsis
- colour terms
- meta-discourse markers
- **formulaic language**

# What types of automatic annotation exist?

1. Part of Speech (POS) tagging
2. Syntactic parsing
3. Semantic annotation

- (6) a. The\_Z5 ending\_T2- of\_Z5 the\_Z5 poem\_Q3 may\_A7+ seem\_A8 to\_Z5  
be\_A3+ contradictory\_A6.1- because\_Z5/A2.2 both\_N5 girls\_S2.1f  
marry\_S4 and\_Z5 have\_A9+ children\_S2mf/T3- ;\_PUNC thereby\_Z5  
filling\_N5.1+ the\_Z5 traditional\_S1.1.1 female\_S2.1 role\_I3.1 .\_PUNC  
b. at\_T1.1.2[i165.3.1 a\_T1.1.2[i165.3.2 time\_T1.1.2[i165.3.3

(Newman and Cox, 2021: 35)

## Q Your research:

- Spanish **articles**
- **morpho-syntactic** and **syntactic** labels
- **morphology**
- **semantic**, **syntactic** and **discourse** features
- **verb valency patterns**
- **grammatical complexity**
- grammatical **case**
- cohesion and **cohesive devices**
- **stance features**
- **verb aspect**
- **adverbs** of degree and negation
- **verb phrase** ellipsis
- **colour terms**
- **meta-discourse markers**
- **formulaic language**

# What types of automatic annotation exist?

1. Part of Speech (POS) tagging
2. Syntactic parsing
3. Semantic annotation



# What are the advantages of automatic annotation?

- 
- 

1. speed
2. reliability



# Today's session

- Text-preprocessing
- POS-tagging and lemmatization
- Hands on activity: *POS-tagging and lemmatization*
- Syntactic annotation

## *Coffee break*

- Hands on activity: *Syntactic annotation*
- Accuracy measures
- Hands on activity: *Calculating accuracy measures*
- Final remarks





# How this workshop will work



Option 1. Webtools/Excel

Option 2. R

# Text-preprocessing

# What do you notice about this text?

I agree that successful people try <e>news</e> things and take risk rather than only doing what they already know how to do well,for these reasons; By trying new things allow you to be curious to know how someone did it and you will find out how to do it too, that way it make you make a research. By trying new things allow you to be positive in your mind and to have a great desire to succed no matter how difficult is the situation. By trying new things you no that you should be openminded,go through disscussion with people who have done the same thing to learn their ways of doing thing, you should meet or have conversation with a lot of these people in other to learn from their experiences. By trying new things you take a big risks, like in " french we say if you don 't risk you don't have anything",risk in a goog way to takle something. We never know if we might succed or not the only way to do it is to risk.Since we do not loose anything when we risk.ed As for me doing the same thing every day become boring,i can say is a waste of energy and time, To conclude,people who succed try new things and take risks rather than only doing what they already know how to do well.

⚠ Ignore spelling mistakes for the time being...



# What do you notice about this text?

I agree that successful people try `<e>news</e>` things and take risk rather than only doing what they already know how to do `well,for` these reasons; By trying new things allow you to be curious to know how someone did it and you will find out how to do it too, that way it make you make a research. By trying new things allow you to be positive in your mind and to have a great desire to succed no matter how difficult is the situation. By trying new things you no that you should be `openminded,go` through disscussion with people who have done the same thing to learn their ways of doing thing, you should meet or have conversation with a lot of these people in other to learn from their experiences. By trying new things you take a big risks, like in " french we say if you `don't` risk you don't have `anything",risk` in a goog way to takle something. We never know if we might succed or not the only way to do it is to risk.Since we do not loose anything when we risk.ed As for me doing the same thing every day become `boring,i` can say is a waste of energy and time, To `conclude,people` who succed try new things and take risks rather than only doing what they already know how to do well.

(Non-exhaustive) list of things that can cause issues for automatic tools:

- (Inconsistent) file encoding
- Spacing
  - Lack of space between words
  - Unnecessary space between words
  - Double space between words
- 'Stylized' apostrophes or quotation marks
- Accented characters (e.g., à)
- Special characters (e.g., \*, %, |)
- Inconsistent spelling rules (e.g., email/e-mail)
- Coding schemes (e.g., XML)

# What do you notice about this text?

I agree that successful people try `<e>news</e>` things and take risk rather than only doing what they already know how to do `well,for` these reasons; By trying new things allow you to be curious to know how someone did it and you will find out how to do it too, that way it make you make a research. By trying new things allow you to be positive in your mind and to have a great desire to succed no matter how difficult is the situation. By trying new things you no that you should be `openminded,go` through disscussion with people who have done the same thing to learn their ways of doing thing, you should meet or have conversation with a lot of these people in other to learn from their experiences. By trying new things you take a big risks, like in " french we say if you `don't` risk you don't have `anything",risk` in a goog way to takle something. We never know if we might succed or not the only way to do it is to risk.Since we do not loose anything when we risk.ed As for me doing the same thing every day become `boring,i` can say is a waste of energy and time, To `conclude,people` who succed try new things and take risks rather than only doing what they already know how to do well.

(Non-exhaustive) list of things that can cause issues for automatic tools:

- (Inconsistent) file encoding
- Spacing
  - Lack of space between words
  - Unnecessary space between words
  - Double space between words
- 'Stylized' apostrophes or quotation marks
- Accented characters (e.g., à)
- Special characters (e.g., \*, %, |)
- Inconsistent spelling rules (e.g., email/e-mail)
- Coding schemes (e.g., XML)

⚠ Importance of **knowing your corpus...**

# Methods of text cleaning/preprocessing

+ Time intensive

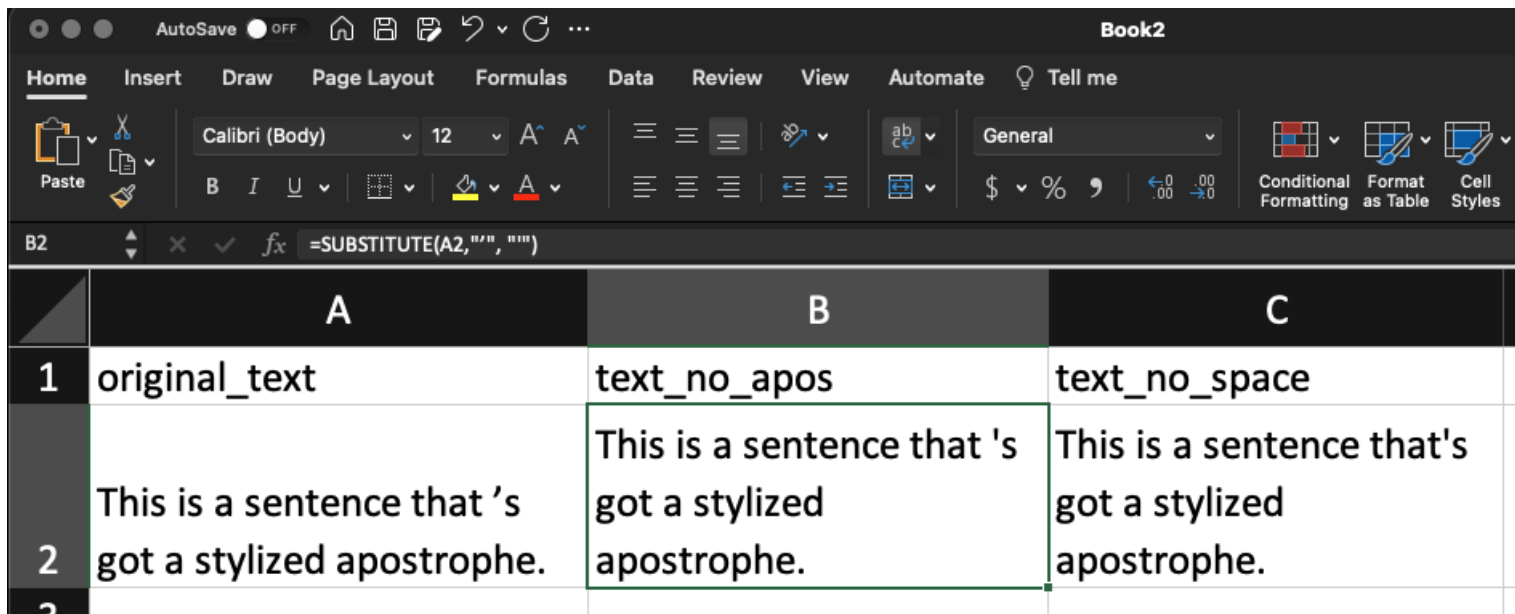
— Replicable

1. Manually
2. Semi-manually using search and replace (e.g., in Excel)
3. Semi-manually using **regular expressions** (e.g., in a text editor)
4. Semi-automatically using a script with regular expressions (e.g., in R or python)

— Time intensive

+ Replicable

# Text pre-processing in Excel



The screenshot shows the Excel interface with the formula bar displaying `=SUBSTITUTE(A2,"'", " ")`. Below the formula bar is a table with three columns: A, B, and C. Row 1 contains headers: 'original\_text', 'text\_no\_apos', and 'text\_no\_space'. Row 2 contains the text 'This is a sentence that's got a stylized apostrophe.' in column A, and its processed versions in columns B and C. The processed text in columns B and C is 'This is a sentence that's got a stylized apostrophe.', where the stylized apostrophe has been replaced with a space.

	A	B	C
1	original_text	text_no_apos	text_no_space
2	This is a sentence that's got a stylized apostrophe.	This is a sentence that's got a stylized apostrophe.	This is a sentence that's got a stylized apostrophe.

SUBSTITUTE() function (For more details see [here](#))

```
=SUBSTITUTE(text, old_text, new_text, [instance_num])  
=SUBSTITUTE(A2,"'", " ")
```

# Text pre-processing in R

```
library(stringr); library(dplyr)
```

```
text <- "This is a sentence that 's got a stylized apostrophe."
```

```
text %>%
```

```
  # replace stylized apostrophes
```

```
  str_replace_all("'", "'") %>%
```

```
  # remove spaces before apostrophes
```

```
  # if they are:
```

```
  # - preceded by the beginning of a string, space or final punctuation
```

```
  # - followed by only one or two letters (e.g., don't, they're)
```

```
  # - and then followed by either a space, final punctuation or end of string
```

```
  str_replace_all("[\\s\\.\\?\\!^]'([^[\\s]{1,2}[\\s\\.\\?\\!$])", "'\\1")
```

```
## [1] "This is a sentence that's got a stylized apostrophe."
```





# With both approaches

⚠ Test (and re-test) your pre-processing pipeline!

- Be careful of inadvertent changes (especially when using regular expressions)  
E.g. *don't* vs. *he said 'don't worry'*
- Be aware the different tools require different approaches (e.g., contractions separate or apart)
- Never edit the original corpus files!
- Keep track of any changes you make

```
text <- "'He 'll be comin' round the 'mountain' when he comes,' I said.'"
text %>%
  str_replace_all("'", "'") %>%
  str_replace_all("[\\s\\.\\?\\!^]'([\\s]{1,2}[\\s\\.\\?\\!$])", "'\\1")
```

```
## [1] "'He'll be comin' round the 'mountain' when he comes,' I said.'"
```



# So why is pre-processing so important?

[1]	"I"	"agree"
[3]	"that"	"successful"
[5]	"people"	"try"
[7]	"&lt;e&gt;news&lt;/e&gt;"	"things"
[9]	"and"	"take"
[11]	"risk"	"rather"
[13]	"than"	"only"
[15]	"doing"	"what"
[17]	"they"	"already"
[19]	"know"	"how"
[21]	"to"	"do"
[23]	"well, for"	"these"
[25]	"reasons;"	" "
[27]	"By"	"trying"
[29]	"new"	"things"



# Tokenization

**Token:** "the smallest unit of a corpus" (Krause, Lüdeling, Odebrecht, and Zeldes, 2012: 2)

= words, numbers, punctuation marks, quotation marks etc.

= syllable, phoneme, etc...

Easiest method (for English):

- split tokens at spaces
- split sentences at periods (.), exclamation marks (!) or question marks (?)

*What problems do you see with this method?*

Potential problems:

- clitics (isn't, ain't)
- missing whitespace
- periods (etc., U.S.A., fig.)
- ordinal numbers
- multiword expressions (New York-based, 10 000, as well as)
- word-internal punctuation (relationship(s), "Rambo"-type)
- (de)hyphenation (preprocessing vs. pre-processing)
- quoted speech ("You still don't have an accountant?" Ellis said.)
- ideographic languages (e.g., Chinese, Japanese)

**Q** For more information about tokenization see Zeldes (2020) and Schmid (2008).

# POS-tagging and lemmatization

# Overview

Each word in the corpus is 'tagged' (labelled) with information about its grammatical category.

🔧 Under the hood:

1. All tokens with unambiguous POS labels are assigned tags (e.g., on the basis of a dictionary)
2. Contextual features (e.g., surrounding tags, morphological endings) used in a statistical model to predict the tags of ambiguous items

(Kyle, 2021: 6)

She 🏷️ pronoun  
sells 🏷️ verb  
seashells 🏷️ noun  
by 🏷️ preposition  
the 🏷️ determiner  
seashore 🏷️ noun  
. 🏷️ punctuation



# Tagsets for English

- CLAWS (Constituent Likelihood Automatic Word-tagging System)
  - CLAWS 5 = 60 tags
  - CLAWS 7 = 160 tags
- PENN Treebank Tagset
- BNC Tagset
- Universal POS tags

*Different tagsets, (subtly) different theories of grammar*

**Table 2.1** Four tagging solutions for English *rid*

	<i>I am now completely rid of such things</i>	<i>You are well rid of him</i>	<i>I got rid of the rubbish</i>
CLAWS7 tagger <sup>a</sup>	Past participle	Past participle	Past participle
Infogistics <sup>b</sup>	Verb base	Verb base	Past participle
FreeLing <sup>c</sup>	Adjective	Verb base	Past participle
(Brill-based) GoTagger <sup>d</sup>	Adjective	Adjective	Adjective

(Newman and Cox, 2021: 21)

💡 Select the tagset that is right for your data/research question.

# Example

- (1) We\_PPIS2 find\_VV0 that\_CST in\_II fact\_NN1 these\_DD2 people\_NN are\_VBR the\_AT most\_RGT exposed\_JJ to\_II media\_NN not\_XX to\_TO mension\_VVI the\_AT fact\_NN1 that\_CST there\_EX is\_VBZ forever\_RT AIDS\_NP1 awareness\_NN1 campaigns\_NN2 launged\_VVN through\_RP out\_RP the\_AT county\_NN1.\_

(ICLE-TS-NOUN-0005.1)

- (2) We find that in fact these people are the most exposed to media not to mention the fact that there is forever AIDS awareness campaigns launged through out the county.

**Verbs:** launged\_VVD, find\_VV0, is\_VBZ, are\_VBR

**Nouns:** media\_NN, fact\_NN1, AIDS\_NN1, county\_NN1, awareness\_NN1, people\_NN, mension\_NN1, campaigns\_NN2, fact\_NN1

= CLAWS C7 Tagset

(van Rooy, 2015: 80-81)

❓ What does the tag PPIS2 refer to?

❓ What do you notice about the learner errors? (e.g., 'mension', 'is awareness campaigns')



# Types of output: *Horizontal*

We\_PPIS2 find\_VV0 that\_CST in\_II fact\_NN1 these\_DD2 people\_NN are\_VBR the\_AT most\_RGT exposed\_JJ to\_II media\_NN not\_XX to\_II mension\_NN1 the\_AT fact\_NN1 that\_CST there\_EX is\_VBZ forever\_RT AIDS\_NN1 awareness\_NN1 campaigns\_NN2 launged\_VVD through\_RP out\_RP the\_AT county\_NN1 .\_.



# Types of output: (*Pseudo*)-XML

```
<w id="2.1" pos="PPIS2">We</w>
<w id="2.2" pos="VV0">find</w>
<w id="2.3" pos="CST">that</w>
<w id="2.4" pos="II">in</w>
<w id="2.5" pos="NN1">fact</w>
<w id="2.6" pos="DD2">these</w>
<w id="2.7" pos="NN">people</w>
<w id="2.8" pos="VBR">are</w>
<w id="2.9" pos="AT">the</w>
<w id="2.10" pos="RGT">most</w>
<w id="2.11" pos="JJ">exposed</w>
<w id="2.12" pos="II">to</w>
<w id="2.13" pos="NN">media</w>
<w id="2.14" pos="XX">not</w>
<w id="2.15" pos="II">to</w>
<w id="2.16" pos="NN1">mension</w>
<w id="2.17" pos="AT">the</w>
<w id="2.18" pos="NN1">fact</w>
<w id="2.19" pos="CST">that</w>
<w id="2.20" pos="EX">there</w>
```

# Types of output: *Vertical*

##	idx	sntc	token	tag	lttr	wclass
## 1	1	1	We	PP	2	pronoun
## 2	2	1	find	VBP	4	verb
## 3	3	1	that	IN	4	preposition
## 4	4	1	in	IN	2	preposition
## 5	5	1	fact	NN	4	noun
## 6	6	1	these	DT	5	determiner
## 7	7	1	people	NNS	6	noun
## 8	8	1	are	VBP	3	verb
## 9	9	1	the	DT	3	determiner
## 10	10	1	most	RBS	4	adverb
## 11	11	1	exposed	VBN	7	verb
## 12	12	1	to	TO	2	to
## 13	13	1	media	NNS	5	noun
## 14	14	1	not	RB	3	adverb
## 15	15	1	to	TO	2	to
## 16	16	1	mension	NN	7	noun
## 17	17	1	the	DT	3	determiner
## 18	18	1	fact	NN	4	noun
## 19	19	1	that	IN	4	preposition
## 20	20	1	there	EX	5	existential

# Lemmatization

**Lemma:** "a 'base form', which provides a level of abstraction from any inflection that might appear in the original orthographic word."

(Newman and Cox, 2021: 29)

(2) We find that in fact these people are the most exposed to media not to mention the fact that there is forever AIDS awareness campaigns launged through out the county.

we find that in fact these people be the most expose to medium not to mension the fact that there be forever AIDS awareness campaign launged through out the county .



# Example

##	idx	sntc	token	tag	lemma	lttr	wclass
## 1	1	1	We	PP	we	2	pronoun
## 2	2	1	find	VBP	find	4	verb
## 3	3	1	that	IN	that	4	preposition
## 4	4	1	in	IN	in	2	preposition
## 5	5	1	fact	NN	fact	4	noun
## 6	6	1	these	DT	these	5	determiner
## 7	7	1	people	NNS	people	6	noun
## 8	8	1	are	VBP	be	3	verb
## 9	9	1	the	DT	the	3	determiner
## 10	10	1	most	RBS	most	4	adverb
## 11	11	1	exposed	VBN	expose	7	verb
## 12	12	1	to	TO	to	2	to
## 13	13	1	media	NNS	medium	5	noun
## 14	14	1	not	RB	not	3	adverb
## 15	15	1	to	TO	to	2	to
## 16	16	1	mension	NN	<unknown>	7	noun
## 17	17	1	the	DT	the	3	determiner
## 18	18	1	fact	NN	fact	4	noun
## 19	19	1	that	IN	that	4	preposition
## 20	20	1	there	EX	there	5	existential

# Webtools

## TreeTagger

### Online TreeTagger

Annotate your texts with part-of-speech and lemma information using [TreeTagger](#).

Type a text

[Upload a file](#)

Text to process\*

Type your text here or copy/paste it

Language of your text\*

## CLAWS

UCREL API

Free CLAWS web tagger

Our free web tagging service offers access to the latest version of the tagger, CLAWS4, which was used to POS tag c.100 million words of the original [British National Corpus \(BNC1994\)](#), the [BNC2014](#), and all the English corpora in Mark Davies' [BYU corpus server](#). You can choose to have output in either the smaller [C5 tagset](#) or the larger [C7 tagset](#).

[CLAWS POS tagger](#) | [Obtaining a licence](#) | [Tagging service](#)

If you would like to use our free WWW tagger, please complete the form below. You can enter up to 100,000 words of English running text. If you enter more, it will be cut off at the word limit. [Input format guidelines](#) are available. To tag the text you have entered click the button below the form.

Select tagset: ☒ C5 ☐ C7

Select output style: ☒ Horizontal ☐ Vertical ☐ Pseudo-XML

Type (or paste) your text to be tagged into this box.

github.com/nvandeweerd/lcr\_ss\_2023

37 / 53

```
library(koRpus); library(koRpus.lang.en)

file <- "data/example_texts/ICLE-TS-NOUN-0005.1.txt"

treetag(
  file,
  treetagger="manual",
  lang="en",
  TT.options=list(
    # Change this to the location where TreeTagger is installed
    path="/Applications/tree-tagger",
    preset="en"
  ),
  doc_id=basename(file)
)
```

⚠ Note: For this to work, both TreeTagger and the appropriate tagsets must first be installed locally on your computer. See instructions [here](#)

💡 See [this vignette](#) for more information about the **koRpus** package.

# Activity 1: POS-tagging

## Webtools/Excel Option

Open `activity_02_pos-tagging.docx` and follow the instructions.

## R Option

Open `activity_02_pos-tagging.R` and follow the instructions.

🔔 Remember that all materials and slides are available on GitHub.

1. Click on `< > Code`.
2. **Download ZIP** to download all files.



# Activity 1: POS-tagging

## Webtools/Excel Option

Open `activity_02_pos-tagging.docx` and follow the instructions.

## R Option

Open `activity_02_pos-tagging.R` and follow the instructions.

🔔 Remember that all materials and are sides available on [GitHub](#)

1. Click on `< > Code`.
2. Download ZIP to download all files.

## Questions

1. What do you notice about the ICLE texts? What pre-processing steps (if any) might be necessary before using automatic annotation tools?
2. R: What 'special characters' do you notice?
3. R: How many 'words' and 'sentences' does BRFF1065.txt contain?
4. R: What is the average sentence length of BRFF1065.txt?
5. WT: How many tokens were tagged with CLAWS (v5)?
6. WT: What is the tag for the base form of a lexical verb in the C5 tagset?
7. WT: What is the tag for the base form of a lexical verb in the C7 tagset?
8. WT: What is meant by '[VVZ/86] NN2/14'?
9. WT: What does the code '@card@' mean?
10. How many adjectives (JJ) are there in BRFF1065.txt?
11. How many common nouns (NN, NNS) are there in BRFF1065.txt?
12. How should you best deal with unknown lemmas?
13. What are the most frequent verb lemmas in the corpus?



# Syntactic parsing

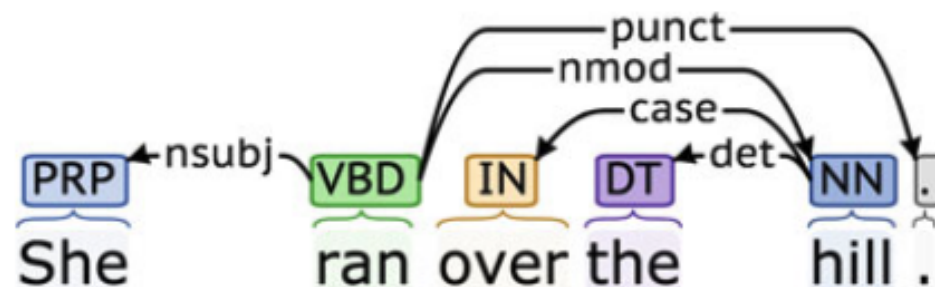
# Overview

Labels of the syntactic connections between words (heads and dependents)

🔧 Under the hood:

1. Texts are POS-tagged.
2. POS tags used in conjunction with phrase-structure rules (generated from training algorithms on large corpora) to generate several possible *parse trees* for each sentence.
3. Statistical or machine learning algorithms are used to select the most probable parse tree for a given sentence.

(Kyle, 2021: 7)



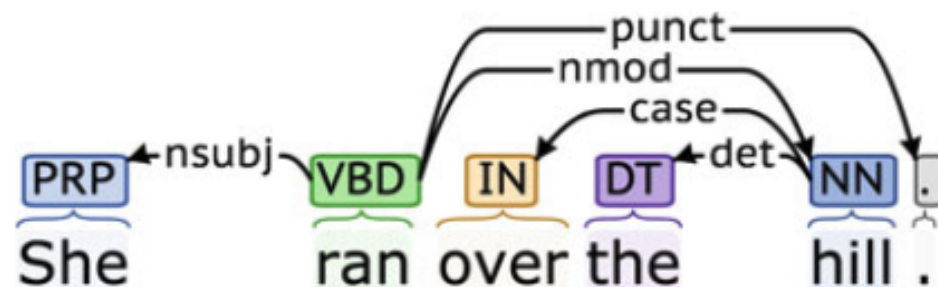
# Overview

Labels of the syntactic connections between words (heads and dependents)

🔧 Under the hood:

1. Texts are POS-tagged.
2. POS tags used in conjunction with phrase-structure rules (generated from training algorithms on large corpora) to generate several possible *parse trees* for each sentence.
3. Statistical or machine learning algorithms are used to select the most probable parse tree for a given sentence.

(Kyle, 2021: 7)



(3) Parse of *She ran over the hill.*

```
(ROOT
  (S
    (NP (PRP She))
    (VP (VBD ran)
      (PP (IN over)
        (NP (DT the) (NN hill))))
    (. .)))
```

# Dependency models for English

- Stanford CoreNLP
- spaCy
- Universal dependencies

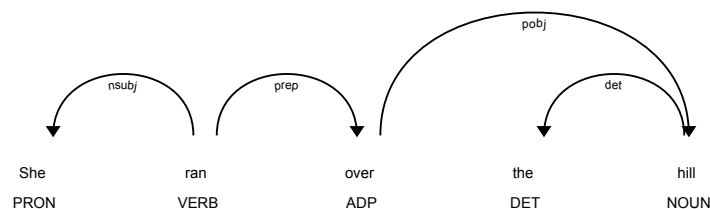
*Different models, (subtly) different theories of grammar*



# Types of output: CoNLL

= *Conference on Natural Language Learning*

	doc_id	sentence_id	token_id	token	lemma	pos	head_token_id	dep_rel
1	text1	1	1	She	she	PRON	2	nsubj
2	text1	1	2	ran	run	VERB	2	ROOT
3	text1	1	3	over	over	ADP	2	prep
4	text1	1	4	the	the	DET	5	det
5	text1	1	5	hill	hill	NOUN	3	pobj
6	text1	1	6	.	.	PUNCT	2	punct



# Types of output: *FoLiA XML*

= *Format for Linguistic Annotation*

```
<p xml:id="example.p.1">
  <t>He hits Mr. Smith. That came quite expected!</t>
  <s xml:id="example.p.1.s.1">
    <t offset="0">He hits Mr. Smith.</t>
    <w xml:id="example.p.1.s.1.w.1"><t offset="0">He</t></w>
    <w xml:id="example.p.1.s.1.w.2"><t offset="3">hits</t>
      <morphology>
        <morpheme class="lexical" function="lexical">
          <t offset="0">hit</t>
        </morpheme>
        <morpheme class="suffix" function="inflectional">
          <t offset="3">s</t>
        </morpheme>
      </morphology>
    </w>
    <w xml:id="example.p.1.s.1.w.3"><t offset="8">Mr.</t></w>
    <w xml:id="example.p.1.s.1.w.4" space="no"><t offset="10">Smith</t></w>
    <w xml:id="example.p.1.s.1.w.5"><t offset="15">.</t></w>
  </s>
  ..
</p>
```

# Types of output: *json*

💡 Can easily be converted into CSV/Excel format.

```
[1] "{"
[2] "  \"text\": \"She ran over the hill\", \"
[3] "  \"ents\": [], \"
[4] "  \"sents\": [ \"
[5] "    { \"
[6] "      \"start\": 0, \"
[7] "      \"end\": 21 \"
[8] "    } \"
[9] "  ], \"
[10] "  \"tokens\": [ \"
[11] "    { \"
[12] "      \"id\": 0, \"
[13] "      \"start\": 0, \"
[14] "      \"end\": 3, \"
[15] "      \"tag\": \"PRP\", \"
[16] "      \"pos\": \"PRON\", \"
[17] "      \"morph\": \"Case=Nom|Gender=Fem|Number=Sing|Person=3|PronType=Prs\", \"
[18] "      \"lemma\": \"she\", \"
[19] "      \"dep\": \"nsubj\", \"
[20] "      \"head\": 1 \"
[21] "    }, \"
[22] "  { \"
[23] "    \"id\": 1 \"
```

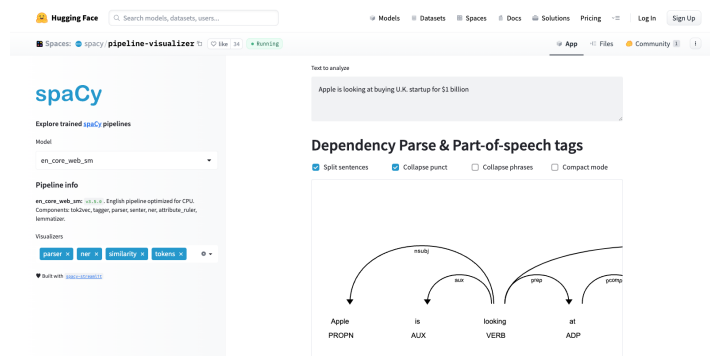
# Webtools

## CoreNLP



⚠ Note: May not be available due to planned outage from June 24th-July 3rd.

## Hugging Face spaCy visualizer





# R

```
library(spacyr)

spacy_initialize(model = "en_core_web_sm")

text <- "She ran over the hill."

spacy_parse(text, dependency = TRUE)
```

⚠ Note: For this to work, spaCy must be installed locally on your computer but this can be done within the spaCy package using the `spacy_install()` function.

💡 See [this vignette](#) for more information about the `spacyr` package.



# Activity 2: Syntactic parsing

## Webtools/Excel Option

Open `activity_02_parsing.docx` and follow the instructions.

## R Option

Open `activity_02_parsing.R` and follow the instructions.

🔔 Remember that all materials and sides available on GitHub.

1. Click on `< > Code`.
2. Download ZIP to download all files.



# Activity 2: Syntactic parsing

## Webtools/Excel Option

Open `activity_02_parsing.docx` and follow the instructions.

## R Option

Open `activity_02_parsing.R` and follow the instructions.

🔔 Remember that all materials and sides available on GitHub.

1. Click on `< > Code`.
2. Download ZIP to download all files.

## Questions

1. Which word is the final period dependent on?
2. What type of dependency relationships are marked by 'amod' and 'dobj'?
3. How many amod dependencies are there in this text?
4. What adjective modifies the word 'difference'?
5. What is the object of the verb 'control'?
6. R: What is the most frequency dependency relation in the corpus?
7. R: What is the average length of noun phrases in the corpus?

Q

& Eh?

L



# Further reading

Kyle, K. (2021). Natural language processing for learner corpus research. *International Journal of Learner Corpus Research*, 7(1), 1–16.

Newman, J., & Cox, C. (2021). Corpus annotation. In *A practical handbook of corpus linguistics* (pp. 25–48). Springer. <https://doi.org/10.4324/9780429269035-7>

Schmid, H. (2008). Tokenizing and part-of-speech tagging. In *Corpus Linguistics: An International Handbook*. de Gruyter.

van Rooy, B. (2015). Annotating learner corpora. In S. Granger, G. Gilquin, & F. Meunier (Eds.), *The Cambridge handbook of learner corpus research* (pp. 79–106). Cambridge University Press. <https://doi.org/10.1017/CBO9781139649414.005>

Zeldes, A. (2020). Corpus Architecture. In M. Paquot & S. Th. Gries (Eds.), *A Practical Handbook of Corpus Linguistics* (pp. 49–73). Springer International Publishing. [https://link.springer.com/10.1007/978-3-030-46216-1\\_3](https://link.springer.com/10.1007/978-3-030-46216-1_3)

