NLP for Geography

Geo-text data mining

Gosse Bouma

Information Science
Groningen University

May 2023

Overview

Amsterdam GPE or Rotterdam GPE ? For sheer picturesqueness, Amsterdam GPE is the easy winner. But what Rotterdam GPE , the Netherlands GPE ' second ORDINAL -largest city, lacks in historical edifices — much of it was bombed in World War II EVENT — it makes up for with contemporary urban cool. Long the busiest port in Europe LOC , the multicultural city is a hub of global commerce and avant-garde architecture. (The architect and Pritzker Prize WORK_OF_ART winner Rem Koolhaas PERSON , a Rotterdam GPE native, has added his touch to the soaring skyline.) Art institutions like the Nederlands Fotomuseum ORG and the new Depot Boijmans Van Beuningen ORG have elevated Rotterdam GPE into an essential European NORP cultural stop, while food markets like the massive, futuristic Markthal FAC and the sleek Foodhallen ORG , which both opened over the past decade DATE , add to a dining scene awash in experimental restaurants.

- Introduction to NLP
- Named Entity Detection and Classification
- Named Entity Linking and Geocoding
- Information Extraction with linguistic patterns
- Using Large Language Models (ChatGPT)

Natural Language Processing

Natural Language Processing

- **Tools** and **applications** for **analyzing** (and generating) **text** and speech
- Very detailed:
  - Models for recognizing the various meanings of the word Python (*word sense disambiguation*)
- Very general:
  - Large Language Models (ChatGPT) as generic building blocks that can be fine-tuned or prompted for specific tasks

Challenges

- *Multilinguality* (English, Dutch, Spanish, Japanese, Hebrew, etc.)
- Register *Variation* (newspaper vs. fiction vs. social media)
- *Ambiguity*: Lexcial (Python), structural (syntactic), named entities (Groningen)

Natural Language Processing

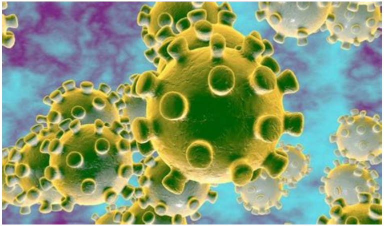
Natural Language Processing

- **Tools** and **applications** for **analyzing** (and generating) **text** and speech
- Very detailed:
  - Models for recognizing the various meanings of the word Python (*word sense disambiguation*)
- Very general:
  - Large Language Models (ChatGPT) as generic building blocks that can be fine-tuned or prompted for specific tasks

Challenges

- *Multilinguality* (English, Dutch, Spanish, Japanese, Hebrew, etc.)
- Register *Variation* (newspaper vs. fiction vs. social media)
- *Ambiguity*: Lexcial (Python), structural (syntactic), named entities (Groningen)

Word Senses: Corona



Word Senses: Python

Word Senses and Embeddings

Word embeddings reflect the *most frequent sense* of a word

```
fasttext nn cc.en.300.bin
```

Python		**Pythons**	
python	0.749	Python	0.641
Pythonic	0.726	pythons	0.619
Python.	0.713	Constrictors	0.565
Perl	0.707	Snakes	0.524
Python-like	0.706	python	0.519
Python3	0.683	Rattlesnakes	0.477
Python-based	0.664	Pythonesque	0.474
Python2	0.659	Monty	0.465
Numpy	0.653	Pythonidae	0.464
Pythons	0.641	Iguanas	0.464

Word Senses and Embeddings

```
fasttext nn cc.en.300.bin
```

Word embeddings reflect the *most frequent sense* of a word

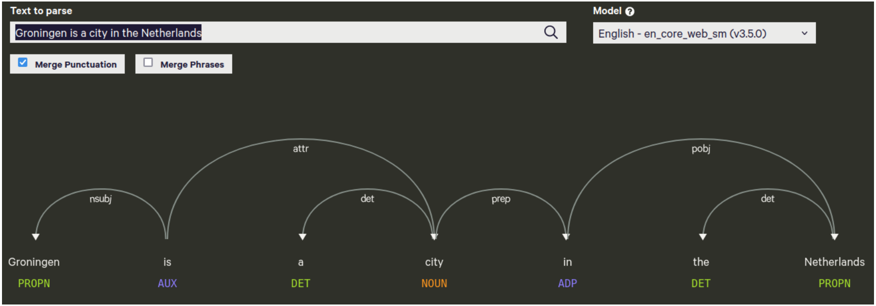
corona		**Corona**	
coronas	0.700	Coronita	0.607
coronae	0.590	Tecate	0.570
Corona	0.531	Hermosa	0.567
aurora	0.493	Coronas	0.567
halo	0.490	Redondo	0.563
chromosphere	0.489	Cerveza	0.557
nanoflares	0.482	Coronado	0.536
filamentary	0.468	corona	0.531
aureole	0.464	Estrella	0.527
halo-like	0.458	Laguna	0.520

Syntactic Ambiguity

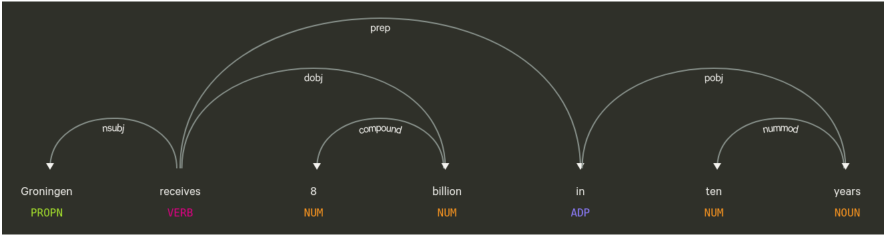
One morning I shot an elephant in my pajamas.
How he got in my pajamas I'll never know." Groucho Marx

Syntactic Ambiguity

One morning I shot an elephant in my pajamas.
How he got in my pajamas I'll never know." Groucho Marx



a city in the Netherlands



receives [8 billion] in ten years

NLP Pipeline

To analyze the linguistic structure of a text involves one or more of the following steps:

- **Preprocessing**: Sentence splitting and tokenization
- **Lexical Analysis**: Lemmatization and Part-of-Speech tagging
- **Syntactc Analysis**: Phrase Structure analysis or Dependency Analysis
- **Semantic Interpretation**: Logical analysis, coreference resolution, word sense disambiguation
- **Discourse Interpretation**: Rhetorical and logical relations between sentences

Natural Language Processing Toolkit

spaCy

spacy.io

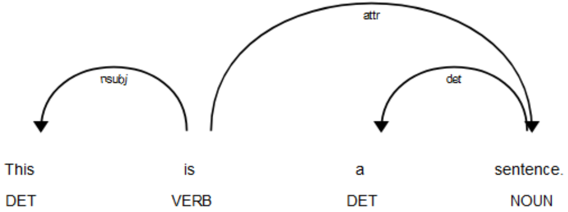
- Python toolkit for analyzing natural language
- Sentence Splitting: segment a text into sentences
- Tokenization: segment a string into a list of tokens
- Lemmatization: label tokens with their lemma (words $\rightarrow$ word, were $\rightarrow$ be)
- Part-of-Speech: label tokens with Part-of-Speech (VERB, NOUN, DET, PROPN, etc.)
- Syntax: syntactic dependency relations between words

Spacy

DEPENDENCY EXAMPLE

```
import spacy
from spacy import displacy

nlp = spacy.load("en_core_web_sm")
doc = nlp("This is a sentence.")
displacy.serve(doc, style="dep")
```



demo : `https://explosion.ai/demos/displacy`

Spacy introduction

```
import spacy
# this loads the model for analysing English text
nlp = spacy.load("en_core_web_sm")

question = nlp('What is the eye color of a siamese cat?')
for word in question :
    print(word.text, word.lemma_, word.pos_)
```

```
What PRON what
is AUX be
the DET the
eye NOUN eye
color NOUN color
of ADP of
a DET a
siamese ADJ siamese
cat NOUN cat
? PUNCT ?
```

Spacy introduction

```
import spacy
# this loads the model for analysing English text
nlp = spacy.load("en_core_web_sm")

question = nlp('What is the eye color of a siamese cat?')
for word in question :
    print(word.text, word.lemma_, word.pos_)
```

```
What PRON what
is AUX be
the DET the
eye NOUN eye
color NOUN color
of ADP of
a DET a
siamese ADJ siamese
cat NOUN cat
? PUNCT ?
```

Syntactic Pattern Matching

Finding Phrases

- Once a text is analysed, we can search for phrases that match a syntactic patterns:
  - Adjective-noun combinations (*'largest city, new model, artificial intelligence'*)
  - Subject-verb-object combinations (*'google-buy-company, koolhaas-win-prize'*)

More Spacy

- Installation: https://spacy.io/usage

```
$ pip install -U pip setuptools wheel
$ pip install -U spacy
$ python -m spacy download nl_core_news_sm
$ python -m spacy download en_core_web_sm
```

- Tutorial: https://spacy.io/usage/spacy-101

Hands-On

Jupyter Notebook

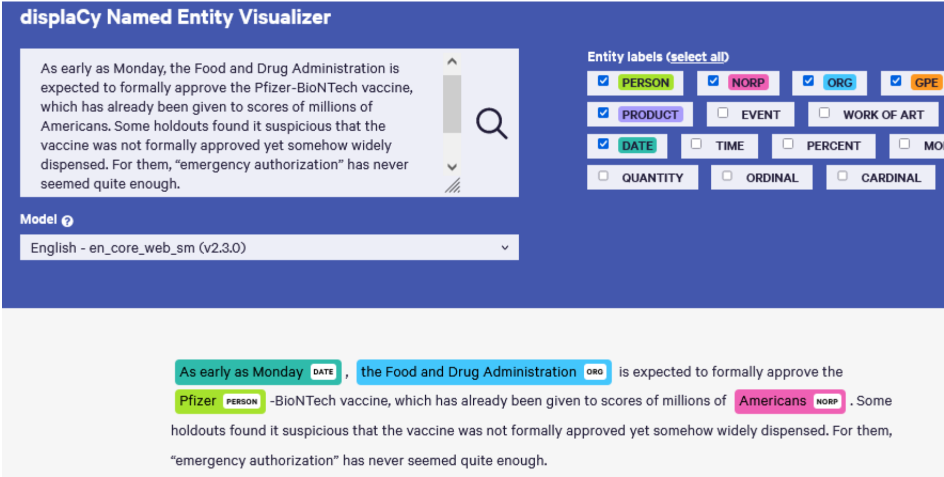
- All resources: `https://github.com/gossebouma/NLP4Geo`
- Download Introduction_to_NLP_with_Spacy.ipynb
- Start `jupyter notebook` in directory where the downloaded file is placed as well
- Open it from the notebook
- **Alternatively:** Go to google colab and upload the notebook file there
- `https://colab.research.google.com/notebooks/welcome.ipynb`

Entity Linking

Sevgili et al., 2020, Neural Entity Linking: A Survey of Models Based on Deep Learning

For instance, imagine a search engine that is able to retrieve mentions in the news during the last month of all retired NBA players with a net income of more than 1 billion US dollars. The list of players together with their income and retirement information may be available in a knowledge base. Equipped with this information, it appears to be straightforward to look up mentions of such retired basketball players in the newswire. However, the main obstacle for such a direct counting algorithm is the lexical ambiguity of entities. In the context of this application, one would want to only retrieve all mentions of Michael Jordan (basketball player) and exclude mentions of other persons with the same name such as Michael Jordan (mathematician).

Entity Linking ≈ NEC + WSD



Named Entity Classification

NEC as a sequence to sequence task

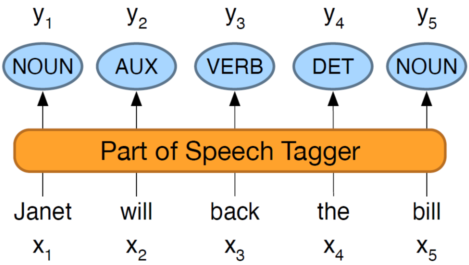
- Use a large pretrained context-sensitive language model to obtain vector representations for words in input
- Sequence-to-sequence labeling task: label each word as belonging to some NE class or as being not a NE
- IOB-labeling scheme:

```
said/O Richard/B-Person Server/I-Person ,/O assistent
director/O of/O
```

Part-of-Speech (PoS) tagging

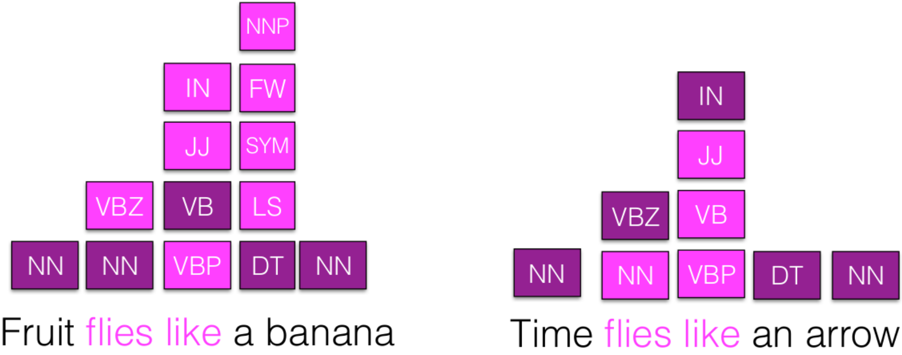
Part-of-Speech Tagging

Map from sequence $x_1,\ldots,x_n$ of words to $y_1,\ldots,y_n$ of POS tags



Ambiguity

Labeling the tag that's correct for the context.



(Just tags in evidence within the Penn Treebank — more are possible!)

How hard is PoS tagging?

- The word bij in Dutch is a preposition (*at, near*), a particle, ..., or a noun (*bee*).
- In the LassySmall corpus *bij* is a preposition 3590 times (99.9%) and a noun 2 times (0.1%)

How hard is PoS tagging?

Most Frequent Class Baseline

A **strong baseline** is always assigning an ambiguous word the PoS-tag that is most frequent for this word in a training corpus.

How To

Train Collect (word,pos-tag) frequencies from a training section of the corpus

Test Label words in test section of the corpus with most frequent PoS-tag, compute accuracy

Performance

Wall Street Journal Corpus: Baseline: approx 92.7% accuracy, state-of-the-art approaches: approx. 97% accuracy

How hard is PoS tagging?

Most Frequent Class Baseline

A **strong baseline** is always assigning an ambiguous word the PoS-tag that is most frequent for this word in a training corpus.

How To

Train Collect (word,pos-tag) frequencies from a training section of the corpus

Test Label words in test section of the corpus with most frequent PoS-tag, compute accuracy

Performance

Wall Street Journal Corpus: Baseline: approx 92.7% accuracy, state-of-the-art approaches: approx. 97% accuracy

How hard is PoS tagging?

Most Frequent Class Baseline

A **strong baseline** is always assigning an ambiguous word the PoS-tag that is most frequent for this word in a training corpus.

How To

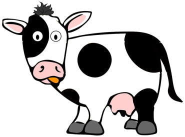
Train Collect (word,pos-tag) frequencies from a training section of the corpus

Test Label words in test section of the corpus with most frequent PoS-tag, compute accuracy

Performance

Wall Street Journal Corpus: Baseline: approx 92.7% accuracy, state-of-the-art approaches: approx. 97% accuracy

State of the Art (Dutch)



Part-of-speech tagging

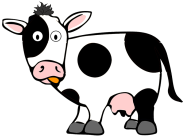
Model	UDv2.5 LassySmall
BERTje	**96.48**
mBERT	96.20
BERT-NL	96.10
RobBERT	95.91

Daniël de Kok
@danieldekok

Excited about the @facebookai XLM-RoBERTa models. The base model beats all other models I have tried so far on Dutch syntax tasks, by quite a large margin (same number of hidden layers and hidden layer size).

Model	POS	Lemma	Morph	LAS
BERT-NL	98.80	98.84	98.81	92.05
Multilingual BERT	98.79	99.04	98.76	92.25
BERTje	98.78	98.79	98.78	92.79
XLM-RoBERTa base	98.94	99.13	98.92	93.16

State of the Art (Dutch)



Part-of-speech tagging

Model	UDv2.5 LassySmall
BERTje	**96.48**
mBERT	96.20
BERT-NL	96.10
RobBERT	95.91

Daniël de Kok
@danieldekok

Excited about the @facebookai XLM-RoBERTa models. The base model beats all other models I have tried so far on Dutch syntax tasks, by quite a large margin (same number of hidden layers and hidden layer size).

Model	POS	Lemma	Morph	LAS
BERT-NL	98.80	98.84	98.81	92.05
Multilingual BERT	98.79	99.04	98.76	92.25
BERTje	98.78	98.79	98.78	92.79
XLM-RoBERTa base	98.94	99.13	98.92	93.16

Neural Models

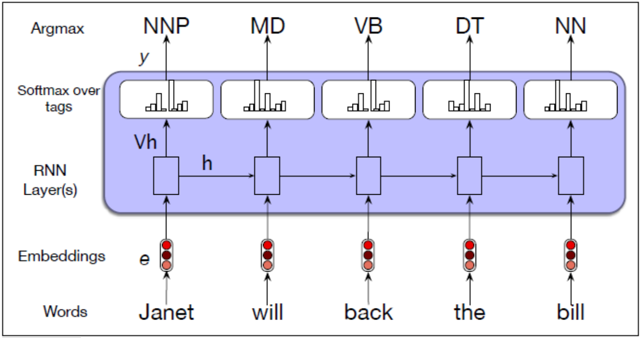


Figure 9.7 Part-of-speech tagging as sequence labeling with a simple RNN. Pre-trained word embeddings serve as inputs and a softmax layer provides a probability distribution over the part-of-speech tags as output at each time step.

Word Sense Disambiguation

Task

- Given a lexicon with M meanings for word W, assign the correct meaning m to occurrence of W in a text

Overview of state-of-the-art

Michele Bevilacqua, et al, *Recent Trends in Word Sense Disambiguation: A Survey,* `https://www.ijcai.org/proceedings/2021/593`
Are knowledge-based methods still relevant? Pure knowledge-based methods are completely outperformed on English WSD... Nevertheless, information within knowledge bases remains valuable and many successful supervised methods are effectively hybridized with knowledge-based methods

WSD – Approaches

Most Frequent Sense baseline

- **Strong baseline**: Assigning the most frequent sense of a word to all its occurrences
- **Drawback**: Requires (lots of) annotated data to obtain reliable frequency estimates
- **Alternative**: Are the nearest neighbors of *python* (using distributional semantics, word embeddings) mostly programming languages or reptiles?

WSD – Approaches

Lesk

- Compute overlap between context of key word in corpus and glosses of its senses

*The bank can guarantee **deposits** will eventually cover future tuition costs because it invests in adjustable-rate **mortgage** securities.*

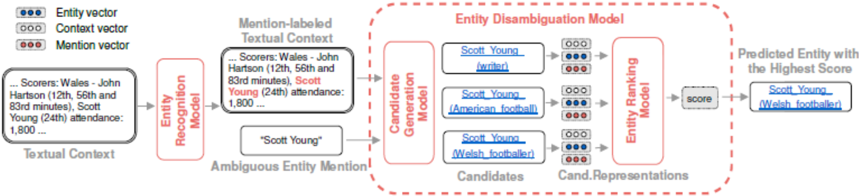
bank[1]	Gloss:	a financial institution that accepts deposits and channels the money into lending activities
	Examples:	"he cashed a check at the bank", "that bank holds the mortgage on my home"
bank[2]	Gloss:	sloping land (especially the slope beside a body of water)
	Examples:	"they pulled the canoe up on the bank", "he sat on the bank of the river and watched the currents"

WSD – Approaches

Contextual Word Embeddings (BERT)

- **Training**:
  - Compute contextual embedding for *bank-1* for each occurrence in manually labeled training data.
  - **Sense embedding** of *bank-1* is the average of all contextual embeddings for *bank-1* in training
- **Testing/Inference**:
  - Compute contextual embedding for some occurrence of *bank* in test-data.
  - Compare with sense embedding for *bank-1*...*bank-n*, choose sense with smallest (cosine) distance.

Neural Entity Linking



Hands-On

Named Entity Classification

- `https://github.com/gossebouma/NLP4Geo`
- Notebook: NamedEntityLinking.ipynb
- Experiment with spaCy Named Entity tagger

Entity Linking

- Notebook: SpacyGeonames.ipynb
- Linking entities to a specific item in Geonames database
- Requires geonames username/password and access to API

Hands-On

Named Entity Classification

- `https://github.com/gossebouma/NLP4Geo`
- Notebook: NamedEntityLinking.ipynb
- Experiment with spaCy Named Entity tagger

Entity Linking

- Notebook: SpacyGeonames.ipynb
- Linking entities to a specific item in Geonames database
- Requires geonames username/password and access to API