sages

Analiza koreferencji

(ang. coreference resolution)

Koreferencja

- To zjawisko **współ odnoszenia się wyrażeń referencyjnych** do tych samych **obiektów**.
- Wyrażenia referencyjne nazywane są **wzmiankami**.
- Grupa wzmianek odnosząca się do tego samego obiektu to **klaster koreferencyjny**.

„**Henryk Sienkiewicz** jest autorem ponad 20 nowel. (...) **Pisarz** jest także autorem powieści oraz reportaży.”

Analiza koreferencji

- Celem **analizy koreferencji** jest połączenie **wyrażeń referencyjnych** (tzw. **wzmianek**) w grupy (tzw. **klastry koreferencyjne**) odnoszących się do tych samych obiektów.
- **Trudność zadania** wynika z faktu, że wzmianki należącego do tego samego klastra **mogą mieć bardzo zróżnicowaną postać** (pod względem formy i kategorii). Wiąże się to m.in. ze względami stylistycznymi oraz praktycznymi.

„Tomek urodził się w Krakowie. Jak miał 16 lat, to jego rodzice zdecydowali się przeprowadzić do Szczecina. Długo nie mógł się pogodzić z ich decyzją o przeprowadzce. Tęsknił za miastem, w którym się urodził."

Typy wzmianek

- fraza rzeczownikowa
  - nazwy własne — Jan Nowak, Warszawa,
  - frazy pospolite — kierownik, stolica,
- zaimek
  - osobowy — on, ona, oni,
  - wskazujący — ten, ta, to, tamten,
  - dzierżawczy — jego, jej, ich,
- podmiot domyślny (Ø)
  - „Mężczyzna szedł po zmierzchu wzdłuż drogi.
    Przejeżdżający samochód oślepił go, przez co Ø **przewrócił** się do rowu.”

Typologia relacji referencyjnych

- relacje bezpośrednie — wzmianka jawnie odnosi się do obiektu,
  - „Wczoraj zacząłem czytać Władcę Pierścieni.
    Ta książka bardzo mnie wciągnęła.”

- relacje pośrednie — wzmianka odnosi się do obiektu poprzez inny obiekt,
  - „Wczoraj zacząłem czytać Władcę Pierścieni.
    Przeczytałem już pierwszy rozdział.”

Zastosowanie analizy koreferencji

- „**Apator wypłaci 1,20 zł** dywidendy i skupi akcje własne za 20 mln zł.”

- „Apator podjął decyzję o wypłacie dywidendy w drugiej połowie 2020 roku. **Spółka wypłaci 1,20 zł** za akcję.”

sages

Koreferencja w spaCy

Pakiet **neuralcoref**

https://github.com/huggingface/neuralcoref

✨NeuralCoref 4.0: Coreference Resolution in spaCy with Neural Networks.

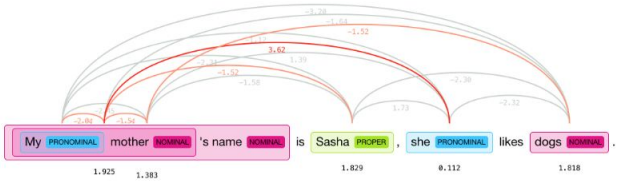
NeuralCoref is a pipeline extension for spaCy 2.1+ which annotates and resolves coreference clusters using a neural network. NeuralCoref is production-ready, integrated in spaCy's NLP pipeline and extensible to new training datasets.

For a brief introduction to coreference resolution and NeuralCoref, please refer to our blog post. NeuralCoref is written in Python/Cython and comes with a pre-trained statistical model for **English only**.

NeuralCoref is accompanied by a visualization client NeuralCoref-Viz, a web interface powered by a REST server that can be tried online. NeuralCoref is released under the MIT license.

✨ Version 4.0 out now! Available on pip and compatible with SpaCy 2.1+.

release v4.0.0 | made with ❤ and spaCy | build passing




```
# Load your usual SpaCy model (one of SpaCy English models)
import spacy
nlp = spacy.load('en')

# Add neural coref to SpaCy's pipe
import neuralcoref
neuralcoref.add_to_pipe(nlp)

# You're done. You can now use NeuralCoref as you usually manipulate a SpaCy document annotations.
doc = nlp(u'My sister has a dog. She loves him.')

doc._.has_coref
doc._.coref_clusters
```

Attribute	Type	Description
`doc._.has_coref`	boolean	Has any coreference has been resolved in the Doc
`doc._.coref_clusters`	list of `Cluster`	All the clusters of corefering mentions in the doc
`doc._.coref_resolved`	unicode	Unicode representation of the doc where each corefering mention is replaced by the main mention in the associated cluster.
`doc._.coref_scores`	Dict of Dict	Scores of the coreference resolution between mentions.
`span._.is_coref`	boolean	Whether the span has at least one corefering mention
`span._.coref_cluster`	`Cluster`	Cluster of mentions that corefer with the span
`span._.coref_scores`	Dict	Scores of the coreference resolution of & span with other mentions (if applicable).
`token._.in_coref`	boolean	Whether the token is inside at least one corefering mention
`token._.coref_clusters`	list of `Cluster`	All the clusters of corefering mentions that contains the token

- **Operating system**: macOS / OS X · Linux · Windows (Cygwin, MinGW, Visual Studio)
- **Python version**: Python 3.6+ (only 64 bit)
- **Package managers**: [pip]

Pakiet **neuralcoref** — parametry

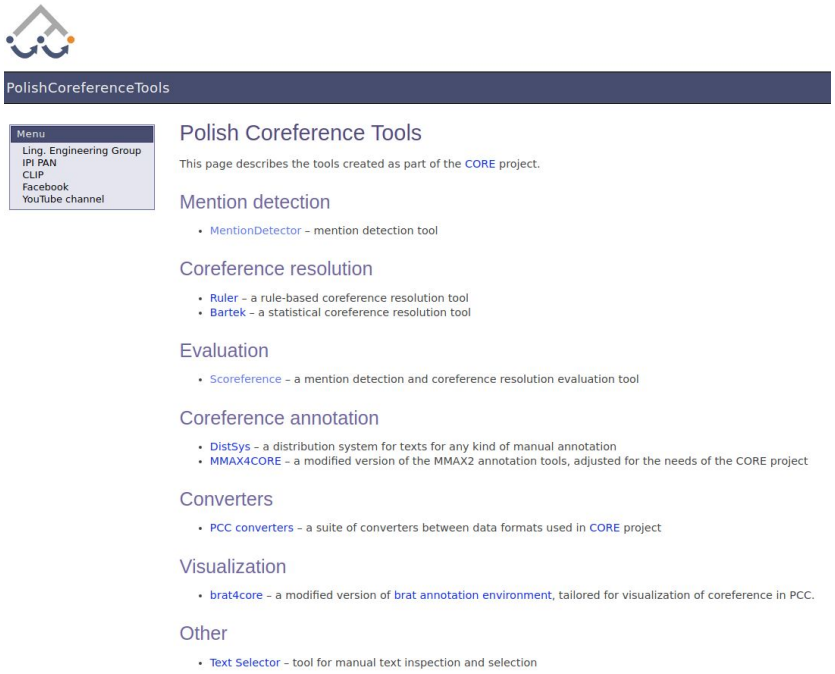
Parameter	Type	Description
`greedyness`	float	A number between 0 and 1 determining how greedy the model is about making coreference decisions (more greedy means more coreference links). The default value is 0.5.
`max_dist`	int	How many mentions back to look when considering possible antecedents of the current mention. Decreasing the value will cause the system to run faster but less accurately. The default value is 50.
`max_dist_match`	int	The system will consider linking the current mention to a preceding one further than `max_dist` away if they share a noun or proper noun. In this case, it looks `max_dist_match` away instead. The default value is 500.
`blacklist`	boolean	Should the system resolve coreferences for pronouns in the following list: `["i", "me", "my", "you", "your"]` . The default value is True (coreference resolved).
`store_scores`	boolean	Should the system store the scores for the coreferences in annotations. The default value is True.
`conv_dict`	dict(str, list(str))	A conversion dictionary that you can use to replace the embeddings of *rare words* (keys) by an average of the embeddings of a list of *common words* (values). Ex: `conv_dict={"Angela": ["woman", "girl"]}` will help resolving coreferences for `Angela` by using the embeddings for the more common `woman` and `girl` instead of the embedding of `Angela` . This currently only works for single words (not for words groups).

```
import spacy
import neuralcoref

nlp = spacy.load('en')

# Let's try before using the conversion dictionary:
neuralcoref.add_to_pipe(nlp)
doc = nlp(u'Deepika has a dog. She loves him. The movie star has always been fond of animals')
doc._.coref_clusters
doc._.coref_resolved
# >>> [Deepika: [Deepika, She, him, The movie star]]
# >>> 'Deepika has a dog. Deepika loves Deepika. Deepika has always been fond of animals'
# >>> Not very good...

# Here are three ways we can add the conversion dictionary
nlp.remove_pipe("neuralcoref")
neuralcoref.add_to_pipe(nlp, conv_dict={'Deepika': ['woman', 'actress']})
# or
nlp.remove_pipe("neuralcoref")
coref = neuralcoref.NeuralCoref(nlp.vocab, conv_dict={'Deepika': ['woman', 'actress']})
nlp.add_pipe(coref, name='neuralcoref')
# or after NeuralCoref is already in SpaCy's pipe, by modifying NeuralCoref in the pipeline
nlp.get_pipe('neuralcoref').set_conv_dict({'Deepika': ['woman', 'actress']})

# Let's try agin with the conversion dictionary:
doc = nlp(u'Deepika has a dog. She loves him. The movie star has always been fond of animals')
doc._.coref_clusters
# >>> [Deepika: [Deepika, She, The movie star], a dog: [a dog, him]]
# >>> 'Deepika has a dog. Deepika loves a dog. Deepika has always been fond of animals'
# >>> A lot better!
```

Koreferencja dla j. polskiego

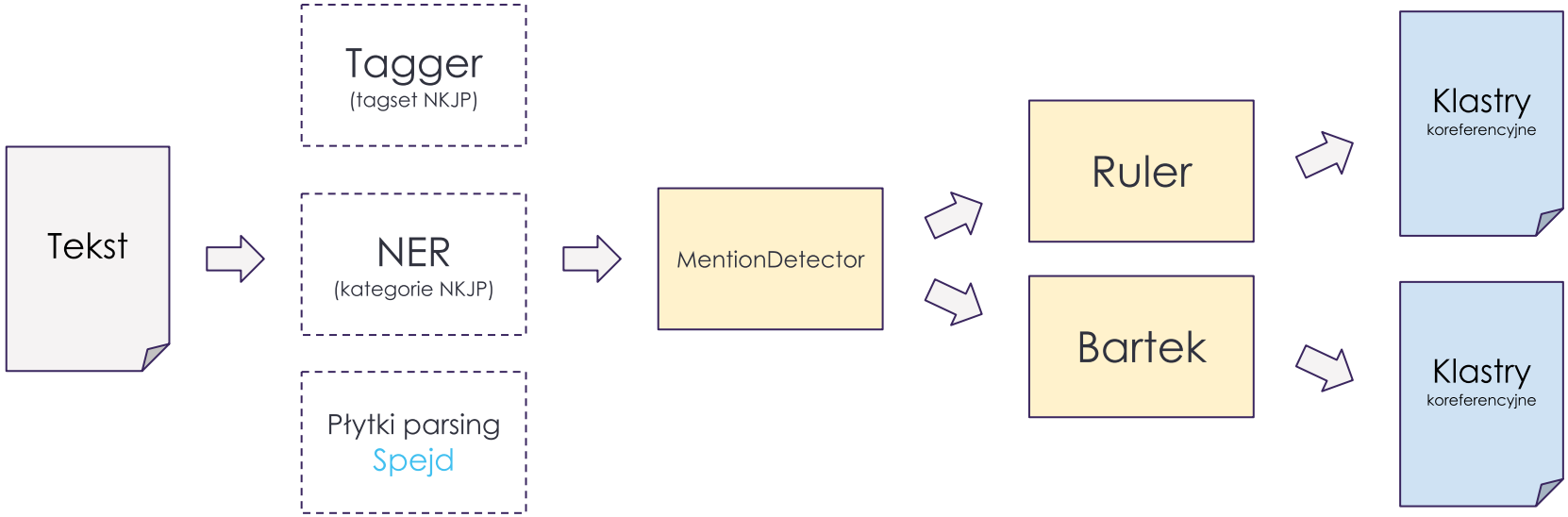
Narzędzia dla j. polskiego

http://zil.ipipan.waw.pl/PolishCoreferenceTools



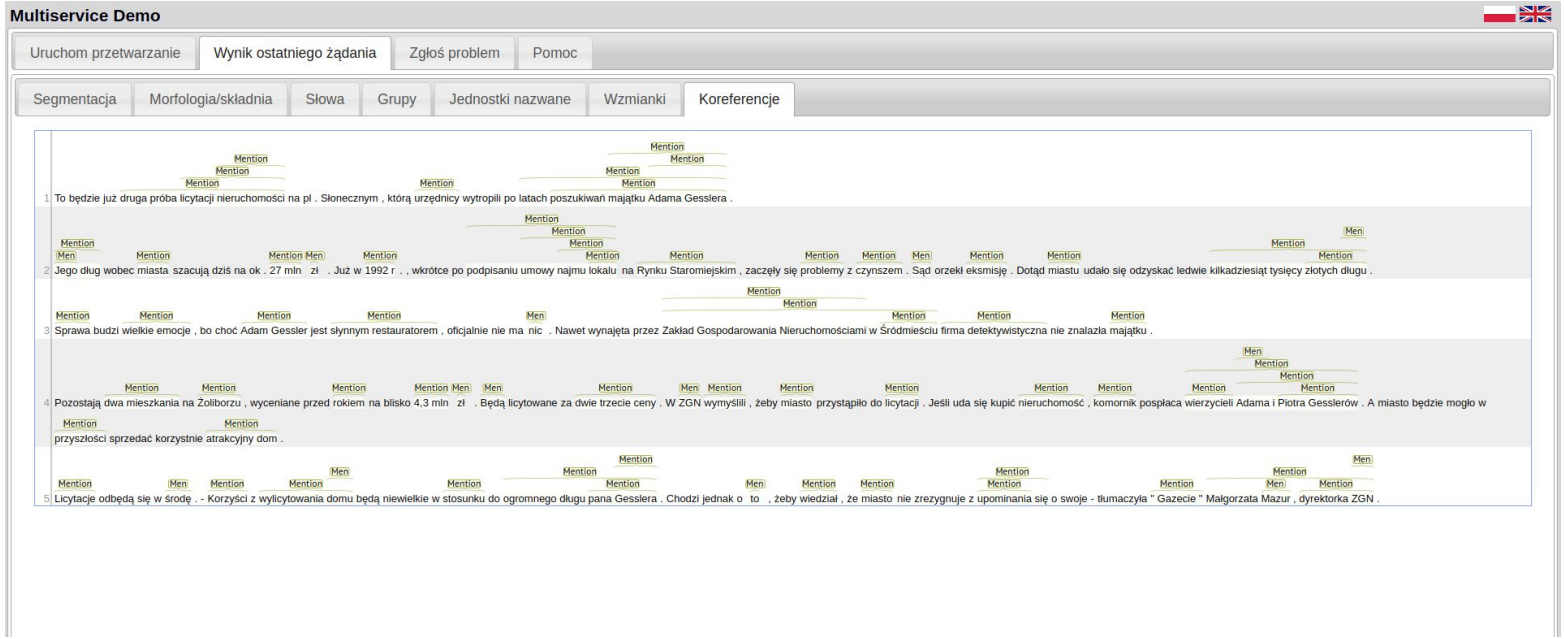
- wzmianki
  - MentionDetector
- klastry koreferencyjne
  - Ruler — regułowy
  - Bartek — statystyczny

Potok przetwarzania



Koreferencja przez Multiservice

http://multiservice.nlp.ipipan.waw.pl



Literatura

Maciej Ogrodniczuk (2019). **Automatyczne wykrywanie nominalnych zależności referencyjnych w polskich tekstach współczesnych**. Wydawnictwo Uniwersytetu Warszawskiego

https://www.wuw.pl/data/include/cms/Automatyczne_wykrywanie_nominalnych_Ogrodniczuk_Maciej_2019.pdf