Semantics Extraction
Technical Report

Group 13
Laudato Lorenzo 0622701563
Mola Sonia 0622701562
Vitolo Alessandra 0622701496

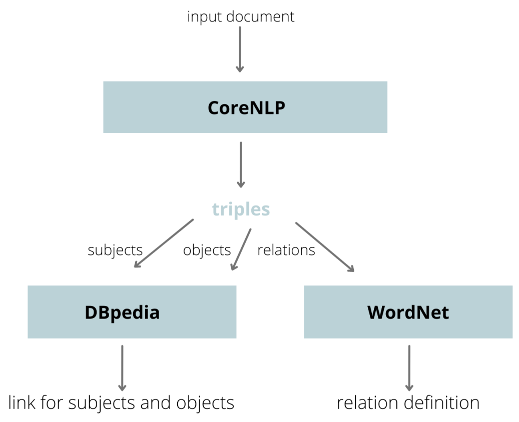
{l.laudato6, s.mola, a.vitolo61}@studenti.unisa.it

Contents

1 Project description

The aim of the project is to extract from text in Natural Language relevant semantic informations.

- **CoreNLP**is used to extract triples *Subject - Relation - Object* from text.
- **WordNet** is used to get the meaning of relations (verbs).
- **DBpedia** is used to extract information from subject and object.



1.1 Requirements

- *NLTK*
- *spaCy*
- *Stanza*
- *CoreNLP*
- *textacy*
- *SPARQLWrapper*

2 Text Preprocessing

2.1 Preprocessing before CoreNLP

The input text is read from a file and then preprocessed. The purpose of preprocessing is to remove *special characters* (hyphens, brackets, etc ..) in order to allow a more precise analysis.

2.2 Preprocessing before DBpedia

DBpedia is interrogated on subjects and objects. These are preprocessed with the aim of obtaining consistent results. In particular, the following checks are carried out:

- If a subject/object is composed of several words, they are separated in order to query DBpedia on the single words. For example, the object *five pizzas* is divided into two entities, *five* and *pizza*, linked to DBpedia with https://dbpedia.org/page/5 and https://dbpedia.org/page/pizza

- If there are terms starting with a lowercase letter, it is converted to uppercase (ex: *pizza* becomes *Pizza*). This is because DBpedia does not recognize entities formatted in this way. For example, https://dbpedia.org/page/pizza in this case the requested entity (*pizza*) in unknown to DBpedia so the query fails, while with https://dbpedia.org/page/Pizza the query is successful and returns the valure of the entity (*Pizza*).
- If a number is expressed in words it is converted into numeric character because in DBpedia the queries on numbers are made on them (ex: *five* becomes *5*).

2.3 Preprocessing before WordNet

WordNet is interrogated on relations. Before searching for the definition of verbs through WordNet, lemmatization is performed on verbs conjugated to obtain their basic form, which is known as the **lemma** (ex: *has become in* in after lemmatizazion is *become*).

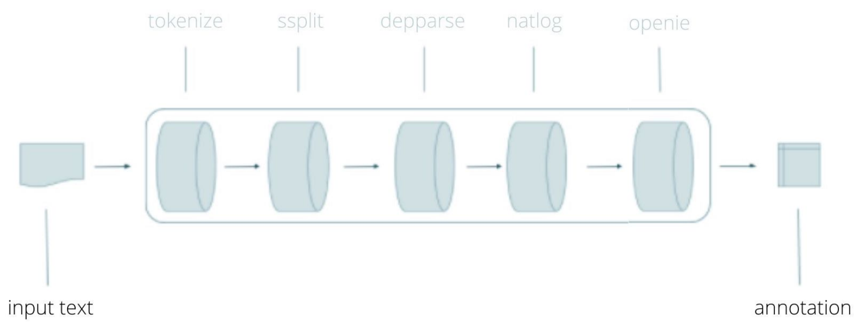
3 System architecture

3.1 CoreNLP for triples extraction

CoreNLP is used to derive linguistic annotations for the text.

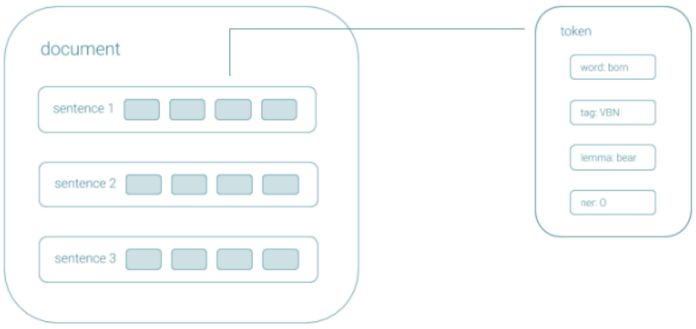
3.1.1 Pipeline

The centerpiece of CoreNLP is the pipeline. Pipelines take in raw text, run a series of NLP annotators on the text, and produce a final set of annotations.



The annotator used are:

- *tokenize*: tokenization is the process of turning text into tokens.
- *ssplit*: sentence splitting is the process of dividing text into sentences.



- *pos*: part of speech tagging assigns part of speech labels to tokens, such as whether they are verbs or nouns.
- *depparse*: provides a fast syntactic dependency parser.
- *natlog*: marks quantifier scope and token polarity, according to natural logic semantics.
- *openie*: the Open Information Extraction (OpenIE) annotator extracts open-domain relation triples, representing a subject, a relation, and the object of the relation.

The output is the triples of the document composed by subject, object and relation.

3.2 WordNet to get Relation Meanings

For each relation (verb) in extracted triples from text it is found the definition of it. From the verb is extract his base form, the lemma. WordNet is queried on it and outputs its definition. For example:

Relation: discover - discover or determine the existence, presence, or fact of

3.3 Query on DBpedia for Subject and Object Linking using SPARQL

The subjects and objects are associated with a DBpedia link that leads to the explanation page of the single entity. SPARQL queries are performed on the individual entities, which returns the link of the entity from DBpedia. For example:

Subject: Harry Potter - http://dbpedia.org/resource/Harry_potter

4 How it works: an example

- **INPUT FILE** :
  "Frodo Baggins knew the Ringwraiths were searching for him - and the Ring of Power he bore that would enable Sauron to destroy all that was good in Middle-earth. Now it was up to Frodo and his faithful servant Sam to carry the Ring to where it could be destroyed - in the very center of Sauron's dark kingdom."
- **INPUT FILE AFTER PREPROCESSING** :
  "Frodo Baggins knew the Ringwraiths were searching for him and the Ring of Power he bore that would enable Sauron to destroy all that was good in Middle earth. Now it was up to Frodo and his faithful servant Sam to carry the Ring to where it could be destroyed in the

very center of Sauron's dark kingdom."

Note that there are no more special characters.

- **OUTPUT OF CORENLP** :

```
{'subject': 'Ringwraiths', 'relation': 'Ring', 'object': 'he
   bore'}
{'subject': 'Sauron', 'relation': 'destroy', 'object': 'all'}
{'subject': 'Ringwraiths', 'relation': 'Ring of', 'object': '
   Power'}
{'subject': 'Ringwraiths', 'relation': 'searching for', 'object
   ': 'him'}
{'subject': 'it', 'relation': 'could', 'object': "could
   destroyed in very center of Sauron 's kingdom"}
{'subject': 'it', 'relation': 'could', 'object': 'where could
   destroyed in very center'}
{'subject': 'it', 'relation': 'carry Ring', 'object': "where
   could destroyed in very center of Sauron 's dark kingdom"}
{'subject': 'it', 'relation': 'could', 'object': "where could
   destroyed in center of Sauron 's dark kingdom"}
{'subject': 'it', 'relation': 'carry Ring', 'object': "could
   destroyed in center of Sauron 's kingdom"}
{'subject': 'it', 'relation': 'carry Ring', 'object': "where
   could destroyed in very center of Sauron 's kingdom"}
{'subject': 'it', 'relation': 'could', 'object': "could
   destroyed in very center of Sauron 's dark kingdom"}
{'subject': 'it', 'relation': 'carry', 'object': 'Ring'}
{'subject': 'it', 'relation': 'could', 'object': "could
   destroyed in center of Sauron 's dark kingdom"}
{'subject': 'it', 'relation': 'could', 'object': 'where could
   destroyed in center'}
{'subject': 'it', 'relation': 'could', 'object': 'could
   destroyed'}
{'subject': 'it', 'relation': 'carry Ring', 'object': 'could
   destroyed in center'}
{'subject': 'it', 'relation': 'could', 'object': "could
   destroyed in center of Sauron 's kingdom"}
{'subject': 'it', 'relation': 'could', 'object': "where could
   destroyed in very center of Sauron 's kingdom"}
{'subject': 'it', 'relation': 'could', 'object': 'could
   destroyed in very center'}
{'subject': 'it', 'relation': 'carry Ring', 'object': "could
   destroyed in very center of Sauron 's dark kingdom"}
{'subject': 'it', 'relation': 'carry Ring', 'object': "could
   destroyed in very center of Sauron 's kingdom"}
{'subject': 'it', 'relation': 'carry Ring', 'object': "where
   could destroyed in center of Sauron 's dark kingdom"}
{'subject': 'it', 'relation': 'could', 'object': "where could
   destroyed in center of Sauron 's kingdom"}
{'subject': 'it', 'relation': 'carry Ring', 'object': "where
   could destroyed in center of Sauron 's kingdom"}
```

```
{'subject': 'it', 'relation': 'could', 'object': 'could
    destroyed in center'}
{'subject': 'it', 'relation': 'carry Ring', 'object': 'could
    destroyed'}
{'subject': 'it', 'relation': 'could', 'object': "where could
    destroyed in very center of Sauron 's dark kingdom"}
{'subject': 'it', 'relation': 'carry Ring', 'object': 'where
    could destroyed in very center'}
{'subject': 'Sauron', 'relation': 'of', 'object': 'dark kingdom
    '}
{'subject': 'it', 'relation': 'could', 'object': 'where could
    destroyed'}
{'subject': 'it', 'relation': 'carry Ring', 'object': 'where
    could destroyed'}
{'subject': 'it', 'relation': 'carry Ring', 'object': 'where
    could destroyed in center'}
{'subject': 'it', 'relation': 'carry Ring', 'object': "could
    destroyed in center of Sauron 's dark kingdom"}
{'subject': 'it', 'relation': 'carry Ring', 'object': 'could
    destroyed in very center'}
```

- **PART OF FINAL OUTPUT** : The final output is saved into the work folder in *.txt* extension.

  Triple n°1 - 'subject': 'Ringwraiths', 'relation': 'Ring', 'object': 'he bore'

  Subject: Ringwraiths - http://dbpedia.org/resource/Ringwraiths

  Object: he bore - ['http://dbpedia.org/resource/He', 'http://dbpedia.org/resource/Bore']

  Triple n°2 - 'subject': 'Sauron', 'relation': 'destroy', 'object': 'all'

  Subject: Sauron - http://dbpedia.org/resource/Sauron

  Relation: destroy - do away with, cause the destruction or undoing of

  Object: all - http://dbpedia.org/resource/All

  Triple n°3 - 'subject': 'Ringwraiths', 'relation': 'Ring of', 'object': 'Power'

  Subject: Ringwraiths - http://dbpedia.org/resource/Ringwraiths

  Object: Power - http://dbpedia.org/resource/Power

  Triple n°4 - 'subject': 'Ringwraiths', 'relation': 'searching for', 'object': 'him'

  Subject: Ringwraiths - http://dbpedia.org/resource/Ringwraiths

  Relation: search - try to locate or discover, or try to establish the existence of

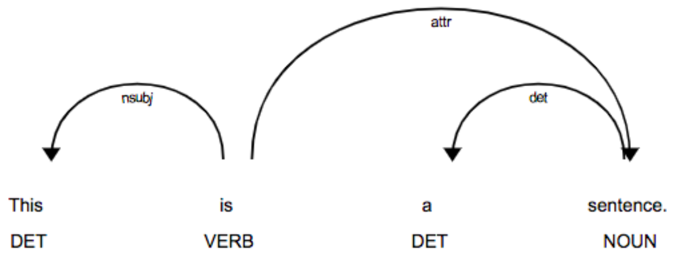
  Object: him - http://dbpedia.org/resource/Him

5 Text Analysis

A further Colab script was devoted to text analysis.

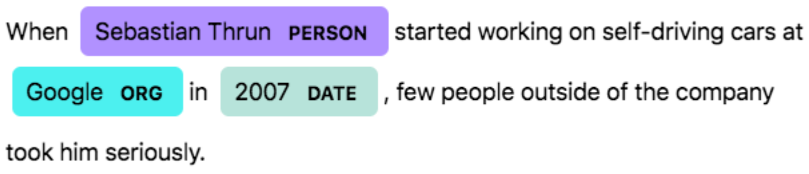
5.1 Visualizing the dependency parse

The *dependency visualizer* is found for each sentence in the document. The dependency visualizer, shows part-of-speech tags and syntactic dependencies.
Example of dependency visualizer:



5.2 Visualizing the entity recognizer

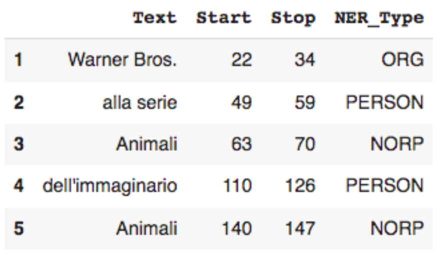
The *entity visualizer* is found for each sentence in the document. The entity visualizer highlights named entities and their labels in a text.
Example of entity visualizer:



When Sebastian Thrun PERSON started working on self-driving cars at Google ORG in 2007 DATE , few people outside of the company took him seriously.

5.3 Word indices

In this section a table is generated which indicates for each word of the sentence where it begins and where it ends.
Example of the table:

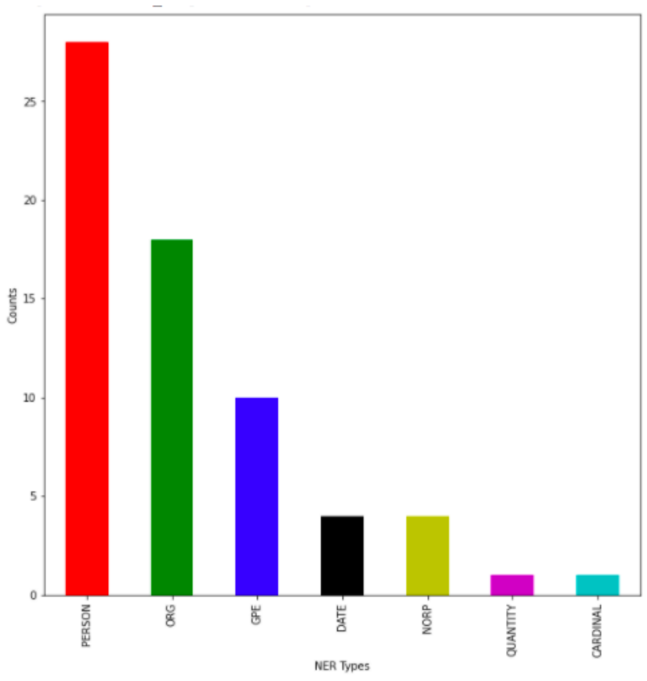


	Text	Start	Stop	NER_Type
1	Warner Bros.	22	34	ORG
2	alla serie	49	59	PERSON
3	Animali	63	70	NORP
4	dell'immaginario	110	126	PERSON
5	Animali	140	147	NORP

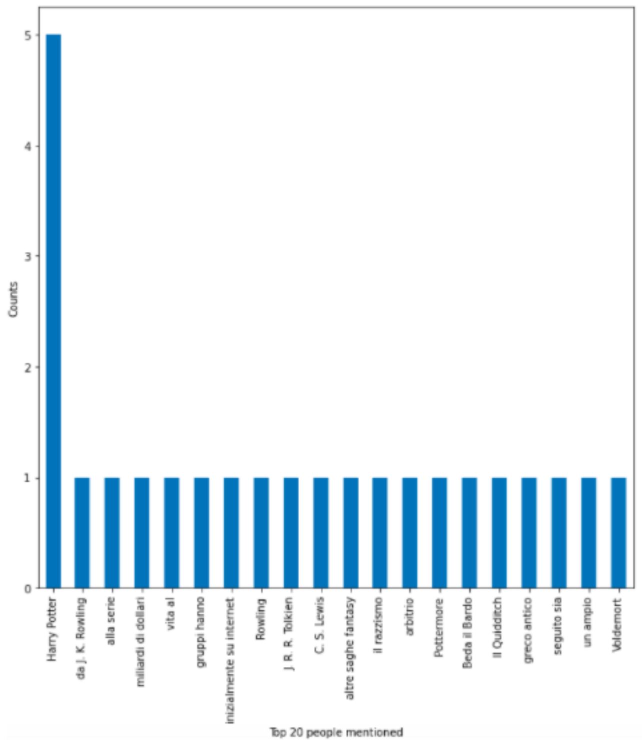
5.4 Named Entity Recognition Histogram

In this section a histogram is generated which represents the number of occurrences for each recognized entity.

Example of the histogram:



A histogram is also generated showing the people most mentioned in the document Example of the histogram:



6 How to run

The source code is located in the folder ' *ProgettoTS*'. It necessary to create a shortcut of this directory in your Google Drive. The Google Colaboratory notebook to run is ' *SemanticsExtraction.ipynb*'. It is divided in four sections that must be performed sequentially.

The four steps are the following:

1. Mount the drive
2. Setup and Requirements
3. Reading the input file
4. Triples extraction

The Google Colaboratory notebook dedicated to the analysis of the text cited in paragraph 5 is' *TextAnalysis.ipynb*'.