SpaCy objektum bővítése

H. Zováthi Örkény

2017. április 22.

Tartalomjegyzék

1. SpaCy objektum bővítése

Az objektum bővítése a Token class Token.doc.user_token_hooks szótárának igénybevételével lehetséges. A szótár kulcsa maga a Token objektum, az érték pedig egy általunk létrehozott **Concept class** lesz, mely a fordításhoz szükséges adatokat tartalmazza.

1.1. Concept class

1.1.1. Belső tagváltozók

- self.mentalese : string
  Tárolja az adott concepthez tartozó pillanatnyi mentalese értéket
- self.relation : Relation(enum)
  A reláció típusát mutatja meg. A lehetséges értékeket következő alfejezetben tárgyalom.
- self.probability : float
  Az a bizonyos p- érték, alapesetben 1.0.
- self.is_processed : bool
  Megmutatja, hogy az adott concept már része- e egy másik conceptnek – azaz mentalese stringje nem lényeges –, vagy az adott pillanatig nem volt beolvasztva egy másik conceptbe se – mentalese string releváns–.
  Ha a változó értéke 1, akkor a concepttel már nincs elvégezendő feladat/ szabály.

1.1.2. Tagfüggvények

- set_mentalese: A reláció függvényében beállítja az adott concept mentalese stringjét.
- update_mentalese: Az adott concept p- értékének függvényében módosítja a mentalese stringet. Amennyiben $p = 1$, nem történik változás.
- set_is_processed: Az adott concepten végzett műveletek után 1-be állítja az *is_processed* belső tagváltozót.
- is_question: Amennyiben a vizsgált concepthez tartozó Token objektum .lemma_ paramétere „?”, *igaz*, különben *hamis* értékkel tér vissza.

1.2. Relációfajták defíniálása

Minden egyes Concepthez tartozik egy relációtípus, ami a következő lehet:

Típus	Rövidítés
Word	'W'
Similar	'S'
Identical	'D'
Class	'C'
Feature	'F'
Determination	'Q'
Action	'A'
Impact	'I'
Relative	'R'
Time	'T'
Part	'P'
More	'M'
Implication	'IM'
NecessaryCondition	'N'
Relevance	'V'
And	'AND'
Or	'OR'
Not	'NOT'
MergeConcept	'MC'

MergeConcept bevezetése:
A bővítést az a szabály tette szükségessé, amely az azonos mondatbeli szerepet belöltő concepteket gyúrja össze egy összetett conceptté.
Ebben az esetben a létrejövő fogalom mentalese stringje az eredeti két mentalese vesszővel elválasztott megfelelője lesz.
Azaz: $m = m1, m2$
Példa:
/e Big red foxes hide usually easily. /m F(A(F(fox,big,red),hide),usually,easily)
Input: *m1 = big, m2 = red*
Output: *m = big,red*

Getting Started with spaCy

Örkeny H. Zovathi

April 21, 2017

Contents

1 Install spaCy

spaCy is compatible with **64-bit CPython 2.6+/3.3+** and runs on **Unix/Linux**, **macOS/OS X** and **Windows**. The latest spaCy releases are available over *pip* (source packages only) and *conda*. Installation requires a working build environment.

1.1 pip

Recommended, this is the easiest way.
Using pip, spaCy releases are currently only available as source packages. When using pip it is generally recommended to install packages in a *virtualenv* to avoid modifying system state:

```
virtualenv .env
source .env/bin/activate
pip install spacy
```

1.2 conda

You can now install spaCy via *conda-forge*:

```
conda config --add channels conda-forge
conda install spacy
```

1.3 Compile from source

The other way to install spaCy is to clone its *GitHub repository* and build it from source. That is the common way if you want to make changes to the code base. You'll need to make sure that you have a development enviroment consisting of a Python distribution including header files, a compiler, pip, virtualenv and git installed. The compiler part is the trickiest. How to do that depends on your system.

```
# make sure you are using recent pip/virtualenv versions
python -m pip install -U pip virtualenv
git clone https://github.com/explosion/spaCy
cd spaCy

virtualenv .env
source .env/bin/activate
pip install -r requirements.txt
pip install -e .
```

Compared to regular install via pip, requirements.txt additionally installs developer dependencies such as Cython.

2 Working build environments

2.1 Ubuntu

Install system-level dependencies via apt-get:

```
sudo apt-get install build-essential python-dev git
```

2.2 macOS / OS X

Install a recent version of XCode, including the so-called "Command Line Tools". macOS and OS X ship with Python and git preinstalled.

2.3 Windows

Install a version of *Visual Studio Express* that matches the version that was used to compile your Python interpreter. For official distributions these are:

DISTRIBUTION	VERSION
Python 2.7	Visual Studio 2008
Python 3.4	Visual Studio 2010
Python 3.5+	Visual Studio 2015

3 Download language models

After installation you need to download a language model. We only need the english language model.

```
python -m spacy download en

>>> import spacy
>>> nlp = spacy.load('en')
```

Note the download data is about 1GB, and it split by two parts: parser and glove word2vec modes. Then you can test spaCy.

4 Test spaCy

After installing spaCy, you can test it by the Python or iPython interpreter:
First, load and initialize the nlp data and text processor, this can took about one minute.
In [1]: *import spacy*
In [2]: *nlp = spacy.load('en')*
If it gives no error back then spaCy is successfully installed on your device.