ReTeRom/TEPROLIN 1.6: Definirea modulelor software și a serviciilor oferite de proiect; identificarea adaptărilor pentru modulele NLP existente și a modulelor noi necesare

Radu Ion
Institutul de Cercetări pentru Inteligență Artificială „Mihai Drăgănescu”
Academia Română
radu@racai.ro

1. Introducere

În raportul tehnic al Activității 1.5, „Definirea specificațiilor funcționale și arhitecturale ale platformei integrate și configurabile de prelucrare a textelor” din proiectul TEPROLIN, am definit o serie de 13 operații PLN[1] interoperabile care vor fi implementate de platforma de prelucrare a textelor TEPROLIN (vezi raportul menționat pentru definiția fiecărei operații):

1. Normalizarea textelor în limba română
2. Inserarea automată a diacriticelor românești în texte
3. Despărțirea în silabe a cuvintelor
4. Poziționarea accentului
5. Transcriere fonetică
6. Expandarea numerelor în termeni numerali
7. Expandarea abrevierilor
8. Segmentare la nivel de frază
9. Segmentare la nivel de unitate lexicală
10. Adnotare cu etichete morfo-sintactice („POS tagging”)
11. Lematizare

[1] PLN și NLP sunt abrevierile în română și, respectiv, engleză ale aceluiași domeniu: Prelucrarea Limbajului Natural/Natural Language Processing

12. Identificarea constituenților sintactici („Chunking”)
13. Analiza sintactică cu relații de dependență („Dependency parsing”)

În prezentul raport de cercetare, vom inventaria și detalia modulele software existente la partenerii proiectului sau open-source care implementează operațiile enumerate mai sus.

2. DiacriticsRestoration[2] și javaNLP2[3]

Aceste două module Java sunt puse la dispoziția proiectului de Universitatea „Politehnica” din București (UPB) și implementează următoarele operații:

- Inserarea automată a diacriticelor românești în texte: este realizată de metoda `public static void diacritics_process(String wordName)` din clasa `DiacriticsProcess.java` a modulului DiacriticsRestoration (Ivan, 2016). Această metodă trebuie adaptată platformei TEPROLIN pentru a elimina un apel SRILM[4] care încarcă modelul de limbă pentru fiecare utilizare a metodei `diacritics_process`. De asemenea, metoda trebuie să poată lucra și cu fraze, nu numai cu fișiere, așa cum lucrează în prezent.
- Normalizarea textelor în limba română: este realizată de clasa `CharactersNormalizer.java` din modulul javaNLP2 dar care trebuie adaptată să utilizeze diacriticele românești actuale (acum folosește „ş” și „ţ” în loc de „ș” și „ț”).
- Expandarea numerelor în termeni numerali: este realizată de clasa `NumbersHandler.java` din modulul javaNLP2; se poate utiliza în platforma TEPROLIN fără adaptări.
- Expandarea abrevierilor: este realizată de clasa `AbbreviationsReplacer.java` din modulul javaNLP2; se poate utiliza în platforma TEPROLIN fără adaptări.

[2] http://git.speed.pub.ro/iivan/DiacriticsRestoration.git
[3] http://git.speed.pub.ro/common/javaNLP2.git
[4] http://www.speech.sri.com/projects/srilm/

3. Romanian TTS[5]

Acest modul Python (3.6) a fost furnizat proiectului de Universitatea Tehnică din Cluj-Napoca (UTCN) și implementează următoarele operații:

- Despărțirea în silabe a cuvintelor: se realizează prin apelul metodei `tp.create_syll_features_and_predict`.
- Poziționarea accentului: se realizează prin apelul metodei `tp.create_accent_features_and_predict`.
- Transcriere fonetică: se realizează prin apelul metodei `tp.create_phonetic_features_and_predict`.

Toate aceste apeluri se fac pe niște structuri de date care au fost pregătite în prealabil de alte apeluri de metode. Operațiile menționate mai sus sunt implementate în scriptul `text_processing_av.py` iar adaptarea acestui script la platforma TEPROLIN va include studierea conținutului structurilor de date necesare acestor metode astfel încât acesta să poată fi pregătit de platformă.

4. NLP-Cube[6]

NLP-Cube (Boroș et al., 2018) este un modul open-source, scris în Python (3.6) care realizează procesarea textelor la următoarele niveluri:

- Segmentare la nivel de frază: prin apelul metodei `TieredTokenizer.tokenize(str)`
- Segmentare la nivel de unitate lexicală: se face odată cu apelul metodei `TieredTokenizer.tokenize(str)`
- Adnotare cu etichete morfo-sintactice („POS tagging”): prin apelul metodei `BDRNNTagger.tag(seq)`
- Analiza sintactică cu relații de dependență („Dependency parsing”): prin apelul metodei `BDRNNParser.parse_sequences(seq)`

Operațiile de mai sus sunt implementate cu ajutorul unor rețele neuronale recursive care sunt antrenate pe treebank-ul românesc, adnotat cu relații sintactice de dependență universale în

[5] http://www.romaniantts.com/
[6] https://github.com/adobe/NLP-Cube

cadrul proiectului SSPR[7] (Barbu Mititelu et al., 2016), disponibil pe Internet la adresa http://universaldependencies.org/.

Singura îmbunătățire a acestui modul pentru includerea sa în platforma TEPROLIN constă în adăugarea unui lexicon românesc cu leme astfel încât lematizarea învățată automat să intervină doar în cazul în care cuvântul care se lematizează nu se află în lexicon. În cazul în care cuvântul se află în lexicon, împreună cu eticheta morfo-sintactică atribuită, lema se va recupera din acest lexicon.

5. TTL[8]

TTL (Ion, 2007) este un modul Perl (5.14) care implementează aceleași operații de procesare a textelor ca NLP-Cube dar care adaugă o operație suplimentară:

- Identificarea constituenților sintactici („Chunking”): prin apelul metodei `ttl::chunker($$)`

TTL poate fi integrat în platforma TEPROLIN fără alte îmbunătățiri.

6. Adaptarea modulelor PLN pentru platforma TEPROLIN

Având în vedere că cele mai multe operații ale platformei TEPROLIN sunt implementate în Python 3, platforma TEPROLIN va fi scrisă în limbajul de programare Python 3 iar serviciile web REST vor fi oferite de serverul Flask[9].

Modulele care nu sunt scrise în Python vor implementa o interfață care le va permite să-și încarce resursele de care au nevoie și să comunice cu platforma printr-un canal de comunicare de tip „named pipe” (vezi raportul tehnic al Activității 1.5 pentru descrierea arhitecturii platformei).

[7] http://dev.racai.ro/ti/wordpress/index.php/project/
[8] http://ws.racai.ro/ttlws.wsdl
[9] http://flask.pocoo.org/

7. Concluzii

Prezentul raport tehnic inventariază toate modulele PLN existente care implementează operațiile de prelucrare a textelor din platforma TEPROLIN. Pentru fiecare operație, am identificat apelul de funcție/metodă din modulul corespunzător și am descris, dacă a fost cazul, schimbările care trebuie operate pentru ca modulul să poată fi adăugat platformei.

Am identificat de asemenea limbajul de programare în care vom scrie platforma de prelucrare a textelor TEPROLIN (Python 3) și soluția tehnică pentru serviciile web REST (serverul Flask care este scris în Python 3 de asemenea).

Prezentul raport tehnic împreună cu raportul tehnic al Activității 1.5 „Definirea specificațiilor funcționale și arhitecturale ale platformei integrate și configurabile de prelucrare a textelor" ne oferă toate informațiile de care avem nevoie pentru a implementa platforma de prelucrare a textelor TEPROLIN.

Referințe bibliografice

Verginica Barbu Mititelu, Radu Ion, Radu Simionescu, Elena Irimia, Cenel-Augusto Perez. 2016. The Romanian Treebank Annotated According to Universal Dependencies. In Proceedings of The Tenth International Conference on Natural Language Processing (HrTAL2016), Dubrovnik, Croatia, 29 September – 1 October 2016.

Tiberiu Boroș, Ștefan Daniel Dumitrescu and Ruxandra Burtica. 2018. NLP-Cube: End-to-end raw text processing with neural networks. To appear in Proceedings of the CoNLL-2018 Shared Task "Multilingual Parsing from Raw Text to Universal Dependencies", October 31 - November 1, 2018, Brussels, Belgium.

Radu Ion. 2007. Word Sense Disambiguation Methods Applied to English and Romanian. PhD thesis (in Romanian). Romanian Academy, Bucharest.

Andra-Irina Ivan. 2016. RESTAURAREA DE DIACRITICE ÎN FIȘIERE TEXT COMPLEXE. Lucrare de diplomă, Facultatea de Electronică, Telecomunicații și Tehnologia Informației, Universitatea Politehnica București.