2.6 Raport asupra implementării modulelor NLP noi

Radu Ion

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

1 Introducere

Platforma de prelucrare a textelor românești TEPROLIN (Ion, 2018), dezvoltată în proiectul ReTeRom, a fost îmbogățită cu două noi module:

1. Un modul de recunoaștere a entităților denumite (engl. „Named Entity Recognition” sau, pe scurt, NER) care recunoaște denumiri de locuri (localități de diverse mărimi, țări, etc.), nume de persoane (femei și bărbați, prefixate sau nu de formule de adresare cum ar fi „d-na”, „dr. ing.”, etc.) și nume de organizații (instituții cu ar fi de exemplu „Institutul de Cercetări pentru Inteligență Artificială „Mihai Drăgănescu””);
2. Un modul de recunoaștere a terminologiei medicale (domeniul bio-medical), bazat pe o versiune anterioară a aplicației NLP-Cube (Boroș et al., 2018), antrenat pe corpusul românesc MoNERo (Mitrofan et al., 2019), care recunoaște părți anatomice („artera iliacă”), substanțe chimice, boli („diabet”) și proceduri medicale („abordul arterei iliace”).

TEPROLIN se poate acum autoconfigura să detecteze ce algoritmi să ruleze pentru a satisface cererea utilizatorului. De exemplu, modulul NER are nevoie de un anumit algoritm pentru segmentare lexicală și adnotarea cu etichete morfosintactice (TTL) și nu funcționează cu un alt algoritm care îndeplinește aceleași funcții.

În cele ce urmează, vom descrie pe scurt cum au fost integrate aceste două module în TEPROLIN și vom da link-urile către sursele Python 3 care conțin aceste implementări.

2 Modulul de NER

A fost dezvoltat de colegul nostru Vasile Florian Păiș pentru teza sa de doctorat și este bazat parțial pe Stanford NER (Finkel et al. 2005). Este disponibil ca serviciu web la adresa http://89.38.230.23/ner/ner.php și așa a fost integrat în TEPROLIN.

Modulul NER așteaptă fraza de procesat într-un anumit format, lucru realizat de metoda `def _prepareSentences(self, dto)`, vezi Figura 1 de mai jos. Astfel, fiecare frază este reprezentată pe un format cu 5 coloane, începând cu simbolul de start de frază `<s>`, sfârșind cu simbolul de sfârșit de frază `</s>` și, pe fiecare linie, separate de caracterul TAB, avem forma ocurentă, lema, eticheta morfo-sintactică MSD (Tufiș, 1999), primele 2 caractere din această etichetă și eticheta morfosintactică redusă (Tufiș, 1999).

```
def _prepareSentences(self, dto) -> str:
    result = []

    for i in range(dto.getNumberOfSentences()):
        tsent = dto.getSentenceTokens(i)
        result.append("<s>\t<s>\t<s>\t<s>\t<s>")

        for tok in tsent:
            msd = tok.getMSD()
            shortmsd = ""

            if len(msd) >= 2:
                shortmsd = msd[0:2]
            else:
                shortmsd = msd

            result.append(
                tok.getWordForm() + "\t" + \
                tok.getLemma() + "\t" + \
                msd + "\t" + \
                shortmsd + "\t" + \
                tok.getCTAG() \
            )
        # end for tok
        result.append("</s>\t</s>\t</s>\t</s>\t</s>")
    # end for i
    return "\n".join(result)
```

Figura 1: Pregătirea frazei pentru modulul NER

Fraza astfel procesată este trimisă serviciului web NER care adaugă etichetele de entități pentru fiecare unitate lexicală a frazei, de asemenea în format tabular cu 2 coloane: prima conține forma

ocurentă a cuvântului iar cea de-a doua conține eticheta entității sau „O” dacă cuvântul nu face parte dintr-un nume de entitate. Figura 2 prezintă implementarea metodei `def _runApp(self, dto, opNotDone)` din API-ul platformei TEPROLIN pentru acest

```
def _runApp(self, dto, opNotDone):
    if not TeproAlgo.getNamedEntityRecognitionOperName() in opNotDone:
        return dto

    sentences = self._prepareSentences(dto)
    resp = requests.post(NEROps.nerURL, data = {"tokens": sentences})

    if resp.ok:
        nsentences = resp.text.split("\n")
        i = -1
        csentence = []

        for ntok in nsentences:
            if not ntok:
                # Skip empty strings.
                continue

            if ntok.startswith("<s>"):
                i += 1
            elif ntok.startswith("</s>"):
                tsentence = dto.getSentenceTokens(i)

                if len(csentence) == len(tsentence):
                    for j in range(len(tsentence)):
                        if tsentence[j].getWordForm() == csentence[j][0] and \
                            csentence[j][1] != "O":
                            tsentence[j].setNER(csentence[j][1])

                csentence = []
            else:
                parts = ntok.split()
                csentence.append((parts[0], parts[5]))
        # end for ntok
```

modul NER.

Figura 2: Metoda care execută operația NER în TEPROLIN

3 Modulul de recunoaștere a terminologiei bio-medicale

Așa cum am menționat în Introducere, acest modul se bazează pe o versiune anterioară a aplicației NLP-Cube, versiune care a fost încorporată în TEPROLIN, în directorul „bioner” din

rădăcina arborelui de surse. Cu ajutorul metodei `def createApp(self)`, încărcăm toate resursele necesare (word embeddings, modele de NER, etc.) o singură dată la pornirea platformei. Similar cu modulul de NER, avem o metodă care formatează fraza pentru NLP-Cube, `def _prepareSentences(self, dto)` și, cu metoda `def _runApp(self, dto, opNotDone)`, rulăm modulul de recunoaștere a terminologiei bio-medicale și adăugăm adnotările obiectului de tip DTO (engl. „Data Transfer Object") care conține toate prelucrările platformei și care este primit la intrare și modificat de toate metodele platformei. Figura 3 prezintă și aceste detalii de implementare.

```
def _runApp(self, dto, opNotDone):
    if not TeproAlgo.getBiomedicalNamedEntityRecognitionOperName() in opNotDone:
        return dto

    sequences = self._prepareSentences(dto)
    i = 0

    for seq in sequences:
        rez = self._tagger.tag(seq)
        tsent = dto.getSentenceTokens(i)

        # These are equal, but just in case...
        if len(rez) == len(tsent):
            for j in range(len(rez)):
                if tsent[j].getMSD() == rez[j][1]:
                    bnlabel = rez[j][0]

                    if bnlabel != '' and \
                        bnlabel != '_' and bnlabel != '-':
                        tsent[j].setBioNER(bnlabel)
                    # end if bnlabel
                # end if ==
            # end for j
        # end if len
        i += 1
    # end for
    return dto
```

Figura 3: Rularea modulului de recunoaștere a terminologiei bio-medicale

4 Concluzii

Platforma de prelucrare a textelor românești TEPROLIN s-a îmbogățit cu două noi module de recunoaștere a diferitelor tipuri de entități. Codul sursă este stocat pe GitLab, la adresa

https://gitlab.com/raduion/teprolin. Accesul este moderat, cererea se poate trimite la adresa de email a autorului acestui raport.

Referințe bibliografice

Boroș Tiberiu, Dumitrescu Ștefan Daniel și Burtică Ruxandra. 2018. *NLP-Cube: End-to-End Raw Text Processing With Neural Networks*. În Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies, Association for Computational Linguistics. pp. 171--179. October 2018.

Jenny Rose Finkel, Trond Grenager și Christopher Manning. 2005. *Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling*. În Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics (ACL 2005), pp. 363--370.

Ion Radu. 2018. *TEPROLIN: An Extensible, Online Text Preprocessing Platform for Romanian*. În Proceedings of the International Conference on Linguistic Resources and Tools for Processing Romanian Language (ConsILR 2018), November 22-23, 2018, Iași, România.

Maria Mitrofan, Verginica Barbu Mititelu și Grigorina Mitrofan. 2019. *MoNERo: a Biomedical Gold Standard Corpus for the Romanian Language*. În Proceedings of the BioNLP 2019 workshop, pages 71–79Florence, Italy, August 1, 2019. ©2019 Association for Computational Linguistics.

Dan Tufiș. 1999. *Tiered Tagging and Combined Language Models Classifiers*. În TSD '99 Proceedings of the Second International Workshop on Text, Speech and Dialogue, pp 28--33, Springer-Verlag London, ISBN:3-540-66494-7.