UNIVERSITATEA "POLITEHNICA" BUCUREŞTI
FACULTATEA DE AUTOMATICĂ ŞI CALCULATOARE

PLN TOOLKIT PROJECT

Ingineri:
Ana Găinaru
Gabriel Sandu
Ştefan Dumitrescu

Introducere

Acest proiect a fost elaborat in cadrul cursului de Procesare a Limbajelor Naturale si foloseste ca input un fisier XML elaborat cu ajutorul programului ConcertChat, dezvoltat de catre Fraunhofer (puteti gasi mai multe informatii despre el la adresa http://www.ipsi.fraunhofer.de/concert/index_en.shtml?projects/chat) .

Una din utilizarile cele mai interesante ale ConcertChat este folosirea pentru a rezolva probleme in mod colaborativ, iar experimentele din cadrul programului VMT cu studenti au demonstrat ca problemele pot fi rezolvate mai usor in acest mod.

Proiectul PLN Toolkit isi propune sa analizeze outputul unei sesiuni tipice ConcertChat in care participantii au discutat pe mai multe subiecte, generand o discutie tipica purtata pe Internet de aproximativ 600-1000 replici.

In urma analizei acestei sesiuni de chat, se doreste obtinerea unor rezultate prin metode automate, spre deosebire de metodele de adnotare manuala deja cunoscute.

Se vor folosi algoritmi cunoscuti pentru obtinerea textului in forma tokenizata, cu ajutorul deja existentului framework NLTK pentru Python, ce contine si ontologia WordNet.

In continuare vom prezenta particularitatile fiecarui tip de analiza pe care am implementat-o si output-ul acestora.

Formatul XML ConcertChat

Tokenizare

Eliminare Stopwords

Stemming

Lanturi Lexicale

Topic Extraction

Part Of Speech Tagging

Gasirea Coreferintelor

Dezvoltari ulterioare

- Pentru partea de POST se poate imbunatati tagger-ul daca se creeaza niste expresii regulate mai potrivite sesiunilor de chat, pornind de la exemple.
- Coreferintele pot fi analizate prin mai multi algoritmi pentru eliminarea false-positives
- Se poate folosi un corpus de text adnotat pentru algoritmul de clusterizare, lucru ce ar spori precizia

Concluzii

Dupa observarea modului de rulare al majoritatii algoritmilor, am observat ca in majoritatea cazurilor s-a obtinut rezultatul asteptat prin metodele automate. Exista cateva exceptii: POST si Coreferintele nu functioneaza suficient de precis, ceea ce era de asteptat.

Training-ul la POST este foarte important pentru buna evolutie a algoritmului, care depinde de un set mare de propozitii deja tagged din corpusuri existente. Deoarece noi am dorit pastrarea limitelor normale de timp la antrenarea tagger-ului, am folosit doar 500 de replici, ceea ce nu e suficient, dupa cum s-a observat pentru exemple de verbe care sunt marcate ca si substantive comune la plural NNS. Precizia estimata este 89.6%.

Algoritmul de Coreferinte nu este adaptat suficient lucrului pe text provenit din chat intre 4 persoane, deoarece o replica poate referi o alta la distanta de 100 de replici intermediare, lucru ce duce la o crestere exponentiala a timpului petrecut in algoritm.

In plus, algoritmul depinde si de un POST complet corect, cu genurile specificate pentru fiecare substantiv/pronume, si cu feature-urile gasite perfect: in ceea ce priveste clasa semantica, este greu de evaluat in cod daca un cuvant este din clasa timp, animal, om, etc, din cauza ca functiile ce calculeaza similitudinea dau gres pe exemple banale.

Proiectul nostru arata tehnici care se pot imbunatati in viitor.

Bibliografie

1. Virtual Math Teams Project
http://www.ipsi.fraunhofer.de/concert/index_en.shtml?projects/vmt

2. NLTK Book, *Steven Bird, Ewan Klein, Edward Loper*
http://www.nltk.org/book

3. NLTK API Docs
http://nltk.googlecode.com/svn/trunk/doc/api/index.html

4. Python Docs
http://docs.python.org/

5. Clustering Algorithms for Noun Phrase Coreference Resolution, *Roxana Angheluta, Patrick Jeuniaux, Rudradeb Mitra, Marie-Francine Moens*

6. Noun Phrase Coreference as Clustering, *Claire Cardie, Kiri Wagstaff*

7. CorefDraw: A Tool for Annotation and Visualization of Coreference Data, *Stefan Trausan-Matu, Sanda Harabagiu, Razvan Bunescu*

8. Coreference Resolution: A Survey, *Pradheep Elango*