Izvješće

Semeval 2020 Team Green @ TakeLab: Rino Čala, Ivan Križanić, Ante Pušić, Mario Zec

Zadatak

SemEval je međunarodna radionica o semantičkoj evaluaciji, uz koju se vode natjecanja u rješavanju problema te vrste.

Naš tim je za svoj problem uzeo zadatak 4A iz SemEval 2020, čiji je cilj napraviti sustav koji za dani par rečenica na engleskom jeziku bira koja ima smisla, a koja ne.

Po pravilima natjecanja dopušteno je koristiti razvojni set rečenica za treniranje modela i dodatna sredstva kao već trenirane jezične modele, baze znanja itd.

Rješenje

Naše glavno rješenje nalazi se u semeval_new.ipynb Jupyter bilježnici.

Rješenje koristi NLP biblioteku spaCy pri vektorizaciji rečenica i ML biblioteku sci-kit za modele za klasifikaciju. Uz njih za potrebe NLP-a još koristimo module iz NLTK za *stopwords*, tokenizaciju, lematizaciju i pristup WordNetu.

Koristimo nekoliko modela za izgradnju vektora koji koriste sličnost riječi koje se razlikuju između rečenica s riječima u istoj rečenici, s riječima zajedničkim objema rečenicama u paru, sličnosti između subjekta, predikata i objekta itd. Većina tih modela je implementirano s provedbom lematizacije i bez nje-

Model (SVC *rbf*) smo testirali cross-validacijom od 5 iteracija, odnosno iterativno trenirali na $^4/_5$ rečenica i testirali na ostatku, te tako postigli točnost od 71 ± 9%.

Prednosti

- brzina: sa relativno malim brojem atributa (4-6) postiže se točnost od cca 67%, stoga su izrada vektora i samo treniranje i testiranje vrlo kratki

Mane

- model nije dobar s rečenicama čija točnost se oslanja na opće znanje (npr. „He went to Germany/China to see the Alps.“)

Mogućnosti

- osposobiti model za rečenice čija točnost se oslanja na opće znanje (npr. „He went to Germany/China to see the Alps.“)
- novi načini za koristiti WordNet u analizi teksta
- dodatan rad s BERT-om

Rad

Naše prvo rješenje je koristilo NLP biblioteku spaCy za vektorizaciju rečenica i ML biblioteku sci-kit za modele za klasifikaciju.

Za vektorizaciju najuspješniji je bio veliki spaCy model en_core_web_lg.

Vektor je imao dva dijela: kombinirani vektor para rečenica i kombinirana sličnost svakog tokena i njegovog djeteta u rečenici, za obje rečenice u paru.

Za klasifikaciju smo koristili modele SVC s linearnom jezgrom, SVC s *rbf* jezgrom, LogisticRegression, KNeighborsClassifier, GaussianNB, DecisionTreeClassifier i RandomForestClassifier. Među ovim modelima najuspješniji su bili SVC s linearnom jezgrom i LogisticRegression s točnošću od 57.52% i 58%.

Kasnije smo razvili metodu koja iz para rečenica izvlači riječi koje se ne pronalaze u drugoj rečenici, te smo te riječi uspoređivali sa ostatkom rečenice koristeći spaCy similarity() metodu. Nakon implementiranja unakrsne validacije (*cross-validation*) dobili smo vrijednosti u iznosu od cca 65% samo koristeći novu metodu, te smo tada odbacili prethodno korištene metode.

Zatim smo napravili metodu koja iz rečenica izvlači predikat, subjekt i objekt te tada uspoređuje svaki od njih sa ostatkom rečenice, te zatim uzeli prosjek te sličnosti kao atribut za vektor. To je donijelo malo poboljšanje. Nakon toga smo modificirali metodu koja uspoređuje različite riječi u rečenicama u metodu koja uspoređuje riječi koje su zajedničke. Uveli smo i lematizacijski filter koji je pospješio točnost. Također smo koristeći spaCy chunks izbacili iz rečenica sve riječi koje nisu bile vezane za isječke, te tako postigli malo poboljšanje u primjeni metode sa usporedbom različitih riječi u rečenicama.

Koristili smo i Googleov BERT model za vektorizaciju tako da smo u njega učitali parove rečenica i izvuklji zadnjih 10 slojeva te dobivene vektore ubacili u SVM, no rezultat je bio de facto random. Na temelju toga smo zaključili da postoji nepoznata greška u našoj implementaciji BERT-a i nastavili bez korištenja njega.

Za semantičku analizu koristili smo Wordnet – leksičku bazu podataka za engleski jezik. Pomoću njega prilagodili smo metode za izgradnju vektora tako da koriste sličnost prema hiperonimima, ali to nije davalo dobre rezultate. Isto smo iskustvo imali pri prilagođavanju metoda tako da koriste sličnost riječi prema skupu sinonima.

Na rečenicama smo isprobali dvije vrste preinaka: pretvorbu imenovanih entiteta u općenite riječi za njih i pretvorbu zamjenica u odgovarajuće imenice (npr. oni -> osobe). Prva se nije pokazala korisnom dok druga jest.

Uz prepravke na metodama došli smo do točnosti od 71 ± 9% primjenom cross-validacije, a na test setu koji je priložen na gitu imali smo točnost od 77%.

Također smo isprobali ELMo, a to je je duboka kontekstualizirana reprezentacija riječi (vektor) koja se koristi u obradi prirodnog jezika. Njega smo istrenirali na rečenicama i za dobivene vektore iskoristili prediction_probability da dobijemo vjerojatnost je li rečenica točna ili kriva. Uz njega smo dobili točnost od 64%.