Detekcija sarkazma na Twitteru pomoću vektora reči i rekurentnih neuralnih mreža

Aleksa Krsmanović

Pregled

Vrši se predikcija sarkastičnih tvitova na labeliranim podacima sa takmičenja SemEval-2018.
Trening set predstavlja 80%, a validacioni i test skup po 10% ukupnog seta podataka. Izabran je ovaj odnos jer tačnost predikcije drastično opada sa smanjivanjem trening skupa.

Redosled izvršavanja:

1. Predprocesiranje tvitova I
2. Formiranje seta custom made featur-a + predprocesiranje II
3. Kreiranje bag of words modela i transformacija tvitova u vektor
4. Embedding, treniranje mreže, optimizacija i predikcija

Predprocesiranje I

Sastoji se od više koraka:

1. razdvajanje prema velikom slovu i markiranje hastag-ova:
   #thisReallySucks => #this #really #sucks
2. pretvaranje emotikona u tekst i njihvo markiranje
   :(=> >>sad
3. zamena svih URL-ova sa "URL"
   www.sadasda.com => URL
4. uklanjanje svih znakova interpunkcije osim markera + @

Formiranje seta custom made featur-a + predprocesiranje II

Za svaku reč u svakom tvitu formira se 6 obeležja i vrši predprocesiranje istovremeno. Za svaki tvit se kreira matrica. Ovim je dobijeno +5% tačnosti.

flag 0: da li je rec korisničko ime, ako jeste zamenjuje se sa "person"
flag 1: da li je reč URL
flag 2: da li je reč emoji
flag 3: da li je reč deo hashtaga
flag 4: da li je reč uppercase
flag 5: da li ima ponavljajuće samoglasnike u sebi - loveeeee

Nakon ove faze uklanjaju se svi markeri i vrši se transformacija u mala slova.

Kreiranje bag of words modela i transformacija tvitova u vektor

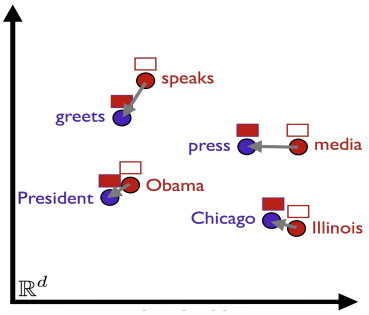
Od dobijenih reči kreira se bag of words model. Svaki tvit se transformiše u vektor čiji su članovi indeksi iz bag-a.
Radi optimizacije treniranja mreže svi tvitovi se zakucavaju na dužinu N članova - ili se dodaju 0 (bag of words indeksi kreću od 1), ili se seče na N.

Embedding, treniranje mreže, optimizacija i predikcija

Na main_input se dovode celi tvitovi - vektor indeksa iz bag of words, a na additional_input se dovode flag vektori za svaku reč u tvitu.
Embedding sloj svaku reč menja sa vektorom od 300 karaktera. Taj vektor omogućava numeričku reprezentaciju koja omogućava da srodne reči imaju slične vrednsoti (*grafik 1*). Koristim GoogleNews-vectors istreniran model.
Nakon embedding sloja, ulazi se spajaju i prosleđuju rekurentnom sloju.
Dropout sloj je ubačen da se spreči overfitovanje koje se dešavalo.

Optimizacija se vrši za vrednost dropout sloja i broja LSTM ćelija u rekurentnom sloju. Zaključeno je da se za vrednosti od 0.1 i 255, respektivno, dobija najveća tačnost na validacionom setu.

Dobijena tačnost na test setu je 69% (f1 mera - 0.7).



grafik 1 - u ovom slučaju se radi o 2D vektorima, a ne o 300D.

Layer (type)	Output shape	Param #	Connected to
main_input (InputLayer)	1,75	0	
embedding_1 (Embedding)	75, 300	2947500	main_input
additional_input (InputLayer)	75, 6	0	
concatenate_1 (Concatenate)	75, 306	0	embedding_1, additional_input
lstm_1 (LSTM)	225	6240	concatenate_1
dropout_1 (Dropout)	225	0	lstm_1
main_output (Dense)	2	12	dropout_1

tabela 1 - arhitektura mreže

Zaključak

Sa mnogo mnogo manje uloženog napora u implementaciju, postignuta je veća tačnost predikcije u odnosu na klasični pristup (66% na istom dataset-u).