Katedra obecné lingvistiky
Univerzita Palackého v Olomouci

Zápočtový projekt z predmetu VMMT2:
Analýza sentimentu filmových recenzií metódami Supervised Machine Learning

Jakub Čieško
2. roč. BŠ
Odbor: FIma-LHmi
2022/2023

1. Popis zadania

Cieľ projektu je spracovať dataset filmových recenzií v angličtine pomocou metód strojového učenia tak, aby výsledný model bol schopný od seba odlíšiť negatívnu a pozitívnu recenziu.

2. Popis dát

Vstupné dáta sa skladajú z jednotlivých textových súborov, ktoré sú rozdelené do troch skupín na základe polarity (negatívne, pozitívne a neutrálne). Celkový počet textových súborov je 3000, pričom každá skupina polarity obsahuje presne 1000 súborov. Priemerná dĺžka pozitívnych recenzií je 104,169 znakov (vrátane medzier), priemerná dĺžka negatívnych recenzií 100,893 znakov (vrátane medzier), neutrálne recenzie sú v priemere dlhé 106,214 znakov (vrátane medzier). Celková priemerná dĺžka recenzií je 103,759 znakov. V našom prípade budeme však pracovať výhradne s pozitívnymi a negatívnymi recenziami, a teda celková veľkosť nášho datasetu bude 2000 recenzií.

3. Preprocessing

Preprocessing je proces predspracovania alebo úpravy dát pred samotnou analýzou alebo modelovaním. V kontexte spracovania prirodzeného jazyka (NLP) sa preprocessing používa na transformáciu a prípravu textových dát pred aplikáciou rôznych algoritmov strojového učenia. Účelom preprocessingu je zlepšiť kvalitu a vhodnosť textových dát pre následné analýzy a modelovanie.

a) Tokenizácia

Tokenizácia je proces rozdelenia textu na jeho jednotlivé časti – tokeny. Tokeny môžu byť slová, interpunkčné znamienka alebo iné významné časti textu. Rozdelenie na tokeny slúži na lepšiu reprezentáciu jednotlivých recenzií, a môže taktiež pomôcť pri hľadaní spojitostí medzi textami, ktoré by v prípade ich celistvosti nebolo možné pozorovať. Výstup tohto procesu taktiež slúži ako vstup pre reprezentáciu textov pomocou Bag of Words. V našom prípade sme tokenizáciu vykonali pomocou knižnice Spacy. Do výsledného zoznamu tokenov sme okrem pár výnimiek („no“, „n't“, „not“) zahrnuli iba tie, ktoré nie sú podľa Spacy modelu en_core_web_trf považované za stop (funkčné) slová. Všetky tokeny sme nakoniec uložili v základnom, lemmatizovanom tvare. Celkový počet tokenov je 4898.

Príklad tokenizácie:

Vstup: `` Simone " is a fun and funky look into an artificial creation in a world that thrives on artificiality .

Výstup: ['thrive', 'fun', 'look', 'artificial', 'creation', 'artificiality', 'world', 'funky']

b) Odstránenie tematických slov

Odstránenie tematických slov spočíva v eliminácii konkrétnych slov z textu, ktoré súvisia s danou témou alebo špecifickou doménou textov. Tieto slová pravdepodobne neposkytujú významnú hodnotu pre následnú analýzu sentimentu. Sú to typicky bežné termíny súvisiace s predmetom textu. V prípade filmových recenzií sa teda môže jednať o slová ako „film“, „movie“, „director“, atď. Zoznam takýchto slov sme vytvorili na základe vlastných znalostí danej domény, ale aj na základe frekvenčného zoznamu tokenov – medzi tematické slová sme zaradili viaceré najčastejšie tokeny. Celkovo sme tak získali zoznam 132 tematických slov, medzi ktoré patria, okrem iných, napríklad tieto: "performance", "hour", "audience", "drama", "touch", "set", "screenplay", "fantasy", "cast", "review", "film", "leave", "day", "care", "adventure".

Nakoľko sa dané slová môžu vyskytovať vo viacerých formách, na ich identifikáciu sme použili SnowballStemmer z knižnice NLTK. Z nášho zoznamu tokenov sme vylúčili všetky tokeny, ktorých tvar po aplikovaní stemmera sa zhoduje s tvarom ktoréhokoľvek tematického slova po aplikovaní stemmera. Takýmto spôsobom sa nám podarilo vylúčiť celkovo 171 typov tokenov.

Príklad odstránenia tematických slov:

Vstup: ['thrive', 'fun', 'look', 'artificial', 'creation', 'artificiality', 'world', 'funky']
Výstup: ['artificial', 'artificiality', 'funky', 'creation', 'thrive', 'fun']

c) Spojenie niektorých foriem prídavných mien a prísloviek

V ďalšom kroku sa snažíme znížiť počet dimenzií slovníka pomocou znalosti gramatiky anglického jazyka. V angličtine existuje niekoľko pravidiel vyjadrujúcich spôsob generovania prídavných mien z prísloviek. Príslovky sú zvyčajne ukončené suffixom „-ly“, v prípade, že tento suffix odstránime, môžeme získať adjektívum, ktoré vyjadruje v podstate to isté, čo daná príslovka. Držíme sa teda prepisovacieho pravidla: ADV(ly) –> ADJ(_ + le + l). V našom súbore tokenov teda nahradíme všetky prípady slov ukončených na „ly“ vhodným prídavným menom. Podmienkou pre náhradu takéhoto adverbia adjektívom je, že dané adjektívum musí už existovať v našej sade tokenov. Touto metódou sa nám podarilo zmenšiť rozmery slovníka o 220 typov tokenov, pričom, myslíme si, sme neprišli o význam daných slov.

Príklad zamenených párov adjektív a adverbií:

"hugely" –> "huge", "lovely" –> "love", "obnoxiously" –> "obnoxious", "hilariously" –> "hilarious"

d) Spojenie niektorých slovies s príponou „-ing“ a ich základných foriem

V tomto kroku, podobne ako v predchádzajúcom, zredukujeme veľkosť slovníka tým, že spojíme niektoré tvary slovies, konkrétne tie, ktoré končia na „-ing“, s ich základnou formou. Použitím obdobného procesu sa nám podarilo zmenšiť slovník o 77 typov tokenov.

Príklad zamenených párov slovies a slovies v „ing“ forme:

"starting" –> "start", "starring" –> "star", "foundering" –> "founder", "thrilling" –> "thrill"

e) Odstránenie potenciálne neexistujúcich slov

V poslednom kroku preprocessingu odstránime potenciálne neexistujúce slová, ktoré mohli vzniknúť ako chyby pri písaní recenzií, a naše výsledky pri určovaní sentimentu by mohli skresľovať. Za takéto chybné slová budeme považovať tie, ktoré sú zložené iba z číslic, alebo obsahujú opakujúcu sa sadu písmen. Touto metódou sa nám podarilo zmenšiť slovník o 19 slov.

Príklad odstránených slov:

"b", "cq", "tv", "aaa", "r", "t", "ai", "xxx", "ed", "zzzzzzzzz", "l", "x", "no", "ol"

4. Vektorizácia

Vektorizácia v NLP slúži na prevod textových dát na numerické vektory, ktoré následne môžu byť spracované algoritmami strojového učenia. V našom prípade použijeme ako reprezentáciu jednotlivých textov 300 rozmerné Word2Vec sémantické embeddingy (word embeddings), ktoré načítame pomocou knižnice Gensim. Texty, pre ktoré vytvárame reprezentáciu, sú syntetické – vznikli popísanými úpravami v preprocessingu.

Hodnotu sémantického embeddingu recenzie stanovíme ako aritmetický priemer hodnôt vektorov jednotlivých slov v recenzii. V prípade, že po našej úprave pri preprocessingu bude existovať recenzia, ktorá neobsahuje nijaké slovo, nahradíme hodnotu vektoru takejto recenzie priemerným vektorom všetkých recenzií rovnakej polarity. Výslednou reprezentáciou nášho problému potom bude trénovacia matica zložená z 1600 riadkov reprezentujúcich jednotlivé texty a 300 stĺpcov reprezentujúcich jednotlivé hodnoty vektora sémantického embeddingu daného textu. Testovacia matica pozostáva zo 400 recenzií a 300 rozmerov sémantických embeddingov.

5. Výsledky jednotlivých metód

Na vyhodnotenie jednotlivých modelov použijeme klasické metriky – precision, recall, accuracy, f1-score – ktoré sú jednoducho získateľné z chybovej matice (confusion matrix). Chybová matica vyjadruje počty správne a nesprávne určených klasifikácií, skladá sa teda zo 4 políčok: TP (true positive), FP (false negative), FN (false negative), TN (true negative). Správne odpovede (TP, TN) sú umiestnené na hlavnej diagonále matice, nesprávne na vedlajšej. Precision sa potom počíta ako podiel TP a súčtu TP s FP. Recall sa počíta ako TP/(TP+FN), accuracy zas ako podiel (TP + TN)/(TP+FP+FN+TN), F1 skóre sa počíta ako 2 * (precision * recall) / (precision + recall).

Nakoľko je takýto výsledok závislý na jedinom pokuse o klasifikáciu, nemusí úplne objektívne vyjadrovať schopnosť modelu zobecňovať, a je preto potrebné model otestovať viacnásobne. Na to nám slúži metóda 10-fold cross validation. Confusion matrix, ako aj spomínané metriky a k-fold cross validation sú implementované v pythone v rámci knižnice scikit-learn, a dajú sa teda jednoducho použiť. V nasledujúcich tabuľkách prezentujeme naše získané výsledky jednotlivých metód: support vector machine (svm), naive bayes, k najbližších susedov (kNN), XGBoost, a LDA.

SVM		
	Trénovací dataset	**Testovací dataset**
Average Precision	0.8514509202453988 (10fold: 0.7542789581356238)	0.9042462311557791 (10fold: 0.7457463118580765)
Recall	0.905 (10fold: 0.82125)	0.93 (10fold: 0.805)
Accuracy	0.895625 (10fold: 0.81375)	0.9325 (10fold: 0.8025)
F1	0.896594427244582 (10fold: 0.8150125095949867)	0.9323308270676693 (10fold: 0.8008411612519444)
Confusion Matrix	[[709 91] [76 724]]	[[187 13] [14 186]]
Stručný popis modelu:		
Sémantické embeddingy; Spacy lemmatizácia; odstránenie kľúčových slov; spojenie niektorých stĺpcov v BoW; prepočítanie chýbajúcich vektorov pomocou priemerného vektoru dobrej a zlej recenzie; kernel: 'rbf', gamma = 0.08, C= 5		

Naive Bayes		
	Trénovací dataset	**Testovací dataset**
Average Precision	0.7138011283497884 (10fold: 0.6940074130209751)	0.7225320512820513 (10fold: 0.6539847847371068)
Recall	0.71 (10fold: 0.68875)	0.655 (10fold: 0.615)
Accuracy	0.766875 (10fold: 0.7474999999999999)	0.765 (10fold: 0.7)
F1	0.7528164347249833 (10fold: 0.7311621709533406)	0.7359550561797754 (10fold: 0.6685556238527376)
Confusion Matrix	[[659 141] [232 568]]	[[175 25]
Stručný popis modelu:		
Sémantické embeddingy; Spacy lemmatizácia; odstránenie kľúčových slov; spojenie niektorých stĺpcov v BoW; prepočítanie chýbajúcich vektorov pomocou priemerného vektoru dobrej a zlej recenzie;		

kNN		
	Trénovací dataset	**Testovací dataset**
Average Precision	0.749700956937799 (10fold: 0.795983913439073)	0.7634343434343434 (10fold: 0.75236303674788)
Recall	0.835 (10fold: 0.7561500203567373)	0.815 (10fold: 0.723631224072737)
Accuracy	0.8125 (10fold: 0.779999)	0.82 (10fold: 0.745)
F1	0.8166259168704156 (10fold: 0.7740425192974648)	0.8190954773869347 (10fold: 0.7360099992806759)
Confusion Matrix	[[632 168] [132 668]]	[[165 35] [37 163]]
Stručný popis modelu:		

Sémantické embeddingy; Spacy lemmatizácia; odstránenie kľúčových slov; spojenie niektorých stĺpcov v BoW; prepočítanie chýbajúcich vektorov pomocou priemerného vektoru dobrej a zlej recenzie; PCA(0.95)
kosínová vzdialenosť;
k_neighbors=11

LDA		
	Trénovací dataset	**Testovací dataset**
Average Precision	0.8214147167487684 (10fold: 0.7988116581353463)	0.9098039215686274 (10fold: 0.7090995021490377)
Recall	0.87875 (10fold: 0.789240860720584)	0.95 (10fold: 0.6952063290054944)
Accuracy	0.87125 (10fold: 0.7943749999999999)	0.94 (10fold: 0.6950000000000001)
F1	0.8722084367245658 (10fold: 0.7926404169437808)	0.9405940594059405 (10fold: 0.6905867054848188)
Confusion Matrix	[[691 109] [97 703]]	[[186 14] [10 190]]
Stručný popis modelu:		

Sémantické embeddingy; Spacy lemmatizácia; odstránenie kľúčových slov; spojenie niektorých stĺpcov v BoW; prepočítanie chýbajúcich vektorov pomocou priemerného vektoru dobrej a zlej recenzie; PCA(0.95)

XGBoost		
	Trénovací dataset	**Testovací dataset**
Average Precision	0.8510748164014688 (10fold: 0.7443140361514963)	0.9875252525252525 (10fold: 0.7108452331981743)
Recall	0.90625 (10fold: 0.749784013456605)	0.985 (10fold: 0.6914718105812465)
Accuracy	0.895625 (10fold: 0.7462500000000001)	0.99 (10fold: 0.705)
F1	0.8967223252937538 (10fold: 0.7457748348750791)	0.9899497487437187 (10fold: 0.6951320248379071)
Confusion Matrix	[[708 92] [75 725]]	[[199 1] [3 197]]
Stručný popis modelu:		

Sémantické embeddingy; Spacy lemmatizácia; odstránenie kľúčových slov; spojenie niektorých stĺpcov v BoW; prepočítanie chýbajúcich vektorov pomocou priemerného vektoru dobrej a zlej recenzie; learning_rate=0.01,max_depth=4, colsample_bytree=0.8

6. Záver

Cieľom projektu bolo spracovať dataset filmových recenzií v angličtine pomocou metód strojového učenia, za účelom vytvoriť model, ktorý by dokázal rozlíšiť medzi pozitívnymi a negatívnymi recenziami. Celkový dataset obsahoval 3000 textových súborov rozdelených do troch skupín podľa polarity: negatívne, pozitívne a neutrálne. Avšak, z dôvodu zamerania sa iba na pozitívne a negatívne recenzie, veľkosť nášho datasetu sa zúžila na 2000 recenzií.

Počas experimentovania s rôznymi metódami strojového učenia sme zistili, že najlepšie výsledky sme dosiahli pomocou metódy SVM (Support Vector Machine). Táto metóda vyžaduje, aby dáta boli lineárne separovateľné. Avšak existujú aj techniky, ktoré dokážu upraviť dáta, ktoré nie sú lineárne separovateľné, tak, aby takými boli.

Nanešťastie s ostatnými metódami sme sa stretli s problémami pri zobecňovaní (generalizácii) – výsledky pri 10fold cross validation neboli dostatočné. Tento problém môže byť spôsobený napríklad nedostatočným alebo chybným preprocessingom alebo samostatnou povahou dát či úlohy.