Tema 3 IA - Sumarizarea documentelor

Vlad Florea

2021

Versiunea 1.1

1 Descrierea problemei

Sumarizarea unui text presupune generarea unei forme restrânse a acestuia păstrând însă informațiile cheie și înțelesul textului inițial, sub forma unui abstract. Procedura este adesea utilizată în aplicații de clasificare a textelor, în sistemele automate de regăsire a răspunsurilor la întrebări, în generarea de abstracte sau titluri pentru știri, etc. Identificăm două moduri în care putem realiza sumarizarea:

- *extractiv* - propozițiile cheie din text sunt identificate și apoi selectate (copiate) pentru a face parte din abstract
- *abstractiv* - textul este interpretat iar abstractul este generat prin metode de prelucrare a limbajului natural astfel încât să descrie în mod fluent și coerent informația importantă

În temă veți implementa soluții de **sumarizare extractivă de știri** în limba engleză.

2 Naive Bayes

Atât clasificarea cât și sumarizarea vor fi efectuate folosind **Naive Bayes**. Aceasta este o metodă statistică inductivă care se bazează pe Teorema lui Bayes, exprimată ca o relație între probabilitate *a priori* și cea *posterioară* a unei ipoteze. Astfel, pentru clasificare avem:

$$P(C = c_k|\mathbf{x}) = \frac{P(\mathbf{x}|C = c_k) \cdot P(C = c_k)}{P(\mathbf{x})} \tag{1}$$

unde:

- $P(C = c_k)$ reprezintă probabilitatea *apriori* a clasei c_K
- $P(C = c_k|\mathbf{x})$ reprezintă probabilitatea *a posteriori* a clasei c_K după ce $\mathbf{x}$ este observat

- $P(\mathbf{x}|C = c_k)$ reprezintă probabilitatea ca $\mathbf{x}$ să facă parte din clasa c_K (*verosimilitate*, eng. *likelihood*)
- $P(\mathbf{x})$ reprezintă probabilitatea observațiilor (eng. *evidence*)

Clasificatoarele Naive Bayes se bazează pe conceptul MAP (eng. *Maximum A Posteriori*), alegând clasa cu probabilitatea maximă:

$$c_{MAP} = \arg\max_{c \in \mathcal{C}} P(\mathbf{x}|c) \cdot P(c) \tag{2}$$

Știm că intrările $\mathbf{x}$ pentru care trebuie să găsim clasa sunt texte formate din N cuvinte (atribute), adică $\mathbf{x} = \{x_1, x_2, ..., x_N\}$. În algoritmul Naive Bayes facem presupunerea simplificatoare prin care considerăm toate atributele x_i ca fiind *condițional independente* când clasa c este observată. Astfel, putem scrie că:

$$P(\mathbf{x}|c) = \prod_{i=1}^{N} P(x_i|c) \tag{3}$$

Astfel, din ecuațiile (2) și (3) avem că:

$$c_{MAP} = \arg\max_{c \in \mathcal{C}} P(c) \cdot \prod_{i=1}^{N} P(x_i|c) \tag{4}$$

iar pentru a evita lucrul cu valori foarte mici care pot duce la erori de calcul (eng. *underflow*), vom logaritma expresia:

$$c_{MAP} = \arg\max_{c \in \mathcal{C}} \log(P(c)) \cdot \sum_{i=1}^{N} \log(P(x_i|c)) \tag{5}$$

3 TF-IDF

TF-IDF (**term frequency–inverse document frequency**) este o metodă numerică statistică ce reflectă cât de reprezentativ este un cuvânt pentru un document dintr-o colecție de mai multe documente.

Pentru **term frequency** vom număra câte apariții are cuvântul in documentul curent:

$$tf(word, d) = freq(word, d) \tag{6}$$

Vom calcula **inverse document frequency** pe baza numărului total de documente din colecție N si a numărului documentelor care conțin cuvântul respectiv $|d \in D : t \in d|$:

$$idf(word, D) = \log(\frac{N}{|d \in D : word \in d|}) \tag{7}$$

Dacă un cuvânt este foarte comun (are apariții in multe documente), scorul său *idf* va tinde spre 0, altfel va tinde spre 1. Dacă înmulțim cei doi termeni vom obține scorul $tf - idf$ pentru cuvântul respectiv:

$$tfidf(word, D) = freq(word, d) \times \log(\frac{N}{|d \in D : word \in d|}) \tag{8}$$

4 Procesări specifice textelor

În vederea obținerii unor rezultate mai bune (și mai relevante) este necesar să efectuăm anumite procesări asupra textelor cu care operăm. Pentru o implementare mai facilă, este recomandat ca în temă să utilizați biblioteci destinate prelucrării în limbaj natural, precum **nltk** sau **spacy**.

Tokenizarea este procesul de spargere a textului în cuvinte [1, 2]. Un proces similar este și cel de împărțire a textului în propoziții (eng. *sentencizer*) [3, 4].

Eliminarea cuvintelor neinformative (eng. *stop-words*) este o tehnică prin care se încearcă reducerea influenței statistice pe care o au anumite cuvinte considerate puțin importante din punct de vedere semantic. Astfel, documentele sunt filtrate pentru eliminarea cuvintelor precum: *to*, *at*, *from*, *and*, *by*, etc.

Lematizarea [5, 6] este procesul prin care toate formele flexionare ale unui cuvânt sunt grupate sub aceeași entitate. De exemplu, în limba engleză, cuvintele *walk*, *walks*, *walked*, *walking*, vor fi toate legate la entitatea *walk*. În acest fel, referențiem împreună forme ușor diferite dar cu semnatică similară crescând relevanța lor statistică.

Utilizarea n-gramelor este o tehnică prin care se dorește realizarea unor calcule statistice care să țină cont și de contextul local al unui cuvânt, dat de cuvintele vecine. Astfel, dacă modelele cu unigrame lucrează cu cuvinte individiuale, modelele cu bigrame utilizează toate perechile de cuvinte adiacente din documente.

5 Sumarizarea documentelor

5.1 Naive Bayes

Având la dispoziție un set de date de antrenare, vom folosi textele din acesta pentru a determina care sunt parametrii modelului nostru. Putem modifica problema de clasificare a textelor studiată în cadrul laboratorului 10, pentru a "clasifica" binar dacă o propozitie face sau nu parte din rezumatul textului. Astfel, obiectul clasificării (intrarea) nu mai este un document, ci o propoziție dintr-un document, iar în loc de cele 5 clase din problema de clasificare în sumarizare ne interesează doar dacă propoziția respectivă va face parte sau nu din abstract. Întrucât și propozițiile sunt formate tot din cuvinte, probabilitățile de verosimilitate se pot calcula similar cu cele din problema clasificării. Vom modifica ecuațiile modelului descrise in 2 astfel:

$$P(SUMMARY) = \frac{\text{număr propoziții care se regăsesc în abstract}}{\text{număr total de propoziții}} \tag{9}$$

$$P(x_i|SUMMARY) = \frac{\text{nr. apariții ale lui } x_i \text{ în propoziții din clasa SUMMARY}}{\text{nr. total cuvinte în propozițiile din clasa SUMMARY}} \tag{10}$$

Din ecuația (10) reiese că pentru cuvinte rare, care apar în textele testate dar nu și în cele cu care antrenăm modelul, probabilitatea de verosimilitate va fi 0, ceea ce va determina $c_{MAP} = 0$ din înmulțirea de termeni (în timp ce $log(0)$ nu este definit). Pentru a evita acest lucru este de dorit ca aceste cuvinte să aibă o valoare diferită de zero, oricât de mică ar fi ea. Prin urmare, vom folosi conceptul de *netezire Laplace* (eng. *Laplace smoothing*):

$$P(x_i|C = c_k) = \frac{\text{nr. apariții ale lui } x_i \text{ în propoziții din clasa SUMMARY } +\alpha}{\text{nr. total cuvinte în propozițiile din clasa SUMMARY } +|Voc| \cdot \alpha} \tag{11}$$

unde α este parametrul de netezire (deseori în practică găsim $\alpha = 1$) iar $|Voc|$ este dimensiunea vocabularului din setul de date.

5.2 TF-IDF

Calculați scorul TF-IDF (Sectiunea 3) pentru fiecare cuvânt din vocabular, pe baza textelor din setul de antrenare. Asigurați-vă că ați preprocesat textul înainte folosind lematizare / stemming (Sectiunea 4).

Puteți calcula un scor pentru fiecare propoziție ca fiind suma valorilor TF-IDF pentru cuvintele din propoziția respectivă. Pentru a îmbunătăți rezultatele obținute urmați următorii pași:

1. Se va calcula scorul doar pentru substantive. Pentru a identifica partile de vorbire puteti folosi resursele din Sectiunea 4 (nltk, spacy, etc.). Adică scorul propoziției va fi afectat doar de scorul substantivelor conținute. Normalizați scorul propoziției.

2. Calculare similaritate cu titlul. Mai exact, se va adăuga o valoare adițională dacă propozitia are cuvinte ce apar în titlu. Această valoare este egală cu numărul de cuvinte din propoziție care se regăsesc în titlu impârțit la numărul total de cuvinte din titlu. La final, această valoare se va pondera cu o constantă care îi definește importanța (de exemplu 0.1) și se va aduna la scorul calculat la pasul 1.

3. La final se va pondera fiecare propoziție cu o pondere (intre 0 și 1) corespunzătoare cu poziția ei în text. De exemplu, pentru un text cu 10 propoziții, ponderea pentru cea de-a noua va fi 0.9. Acest lucru se întamplă

deoarece se tinde ca propozițiile cele mai importante să fie spre finalul articolului (de exemplu concluzia). Definiți propria metrică de ponderare a propoziției în funcție de locația în document.

4. Se vor returna primele k propoziții cu cel mai mare scor, unde k va fi determinat de voi pentru a obține o sumarizare cât mai bună la pasul de evaluare (ex. $k = 3$).

6 Evaluarea performanțelor

Evaluarea performanțelor modelelor se poate face calculând *precizia* și *regăsirea* (eng. *recall*) acestuia [7, 8]. Precizia reprezintă numărul de predicții corecte ale modelului din numărul total de predicții făcute. Valoarea de regăsire a informației se calculează ca fiind numărul de predicții corecte făcute de model pentru o anumită clasă din numărul total de indivizi din clasa respectivă.

Se vor folosi două metrici: $BLEU@N$ și $ROUGE@N$ [9]. În general metrica $BLEU$ măsoară precizia, iar $ROUGE$ recall-ul. Ele sunt definite după cum urmeaza:

$$BLEU@N = \frac{\text{nr. n-grame comune intre text si ground truth}}{\text{nr. n-grame in text}} \tag{12}$$

$$ROUGE@N = \frac{\text{nr. n-grame comune intre text si ground truth}}{\text{nr. n-grame in ground truth}} \tag{13}$$

În comparație se vor folosi metricile pentru unigrame, bigrame si 4-grame, adica (BLEU@1, BLEU@2, BLEU@4 și ROUGE@1, ROUGE@2 și ROUGE@4).

7 Cerințe

Pentru temă veți implementa cele 2 metode de sumarizare bazate pe algoritmii Naive Bayes și TF-IDF. Setul de date cu care veți lucra este *BBC News Summaries*, pe care îl puteți descărca de la adresa: `https://www.kaggle.com/pariza/bbc-news-summary`. Acesta cuprinde știri din 5 categorii (*business*, *entertainment*, *politics*, *sport*, *tech*). Setul de date este împărțit în 2 directoare, unul cu știrile originale iar celălalt cu sumarizările acestora. Setul de date îl veți împărți aleator în două subseturi: unul cu date de antrenare (75%) și unul cu date de testare (25%). Evident, parametrii modelelor vor fi determinați pe baza subsetului de antrenare, în timp ce performanțele le veți raporta bazat pe subsetul de testare.

Cerința 1 (2p) Implementați încărcarea setului de date în memorie, grupând documentele după clasa lor. Fiecărui document trebuie să îi asociați și sumarizarea aferentă. Pentru documentele încărcate aplicați pașii de tokenizare (cuvinte și propoziții), eliminare cuvinte neinformative (eng. *stop-words*) și lematizare.

Pentru acest pas puteti folosi biblioteci existente (nltk, spacy, etc.). Reprezentarea datelor în implementare nu este impusă, însă găsirea unor forme optimizate este încurajată.

Cerința 2 (2p) Implementați algoritmul Naive Bayes pentru sumarizarea știrilor din setul de date. Calculați valorile BLEU@N si ROUGE@N pentru variantele cu sau fără eliminarea *stop-words* (fără lematizare), cu lematizare (dar *stop-words* eliminate). Toate aceste rezultate trebuie calculate pentru unigrame, bigrame si 4-grame.

Cerința 2 (2p) Implementați algoritmul TF-IDF pentru sumarizarea știrilor din setul de date. Calculați valorile BLEU@N si ROUGE@N pentru a evidenția cum influențează scorul fiecare pas din algoritm (cum se comportă algoritmul fără fiecare din cei 3 pași, cum influențează ponderea similarității cu titlul performanța, cum influențează politica de ponderare a propoziției în raport cu locația ei în text performanța). Toate aceste rezultate trebuie calculate pentru unigrame, bigrame si 4-grame.

Cerința 4 (4p) Redactați un raport cu rezultatele experimentale în care să includeți grafice comparative cu rezultatele metodelor testate. Realizați grafice pentru a evidenția diferențele provocate de fiecare modificare adusă celor 2 metode. Realizați un grafic comparativ care ilustrează performanțele tutoror metodelor concomitent. Realizați o analiză calitativă pentru fiecare metodă și variațiile acesteia în parte. Pentru analiza calitativă veți selecta 3 exemple din setul de antrenare si 3 exemple din setul de testare și veți include textul, sumarizarea produsă și cea reală. Formulați explicații despre cum influențează fiecare modificare a celor 2 algoritmi performanța calitativă și cantitativă a acestora. Raportările anterior menționate se vor face și la nivelul fiecărei clase de știri (*business*, *entertainment*, *politics*, *sport*, *tech*).

Bonus (2p) Aduceți îmbunătățiri proprii algoritmului. Punctajul pentru bonus se va acorda în funcție de complexitatea îmbunatățirilor aduse cât și de performanța acestora.

Referințe

[1] `www.nltk.org/api/nltk.tokenize.html`.

[2] `https://spacy.io/api/tokenizer`.

[3] `http://www.nltk.org/api/nltk.tokenize.html#module-nltk.tokenize.punkt`.

[4] `https://spacy.io/api/sentencizer`.

[5] `http://www.nltk.org/api/nltk.stem.html#module-nltk.stem.wordnet`.

[6] `https://spacy.io/api/lemmatizer`.

[7] `https://link.springer.com/referenceworkentry/10.1007%2F978-0-387-30164-8_652`.

[8] https://developers.google.com/machine-learning/crash-course/classification/precision-and-recall.

[9] http://www.ccs.neu.edu/home/vip/teach/DMcourse/5_topicmodel_summ/notes_slides/What-is-ROUGE.pdf.