UNIVERSITÀ DEGLI STUDI FIRENZE

UNIVERSITÀ DEGLI STUDI DI FIRENZE

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

Intelligenza Artificiale

Autore:
Cristian Sician

N° Matricola:
7050925

Corso principale:
Intelligenza Artificiale

Docente corso:
Paolo Frasconi

1 Introduzione

Verrà implementato l'esperimento di Bo Pang et al. 2002 [1], cercando di ottenere risultati analoghi, usando implementazioni di Naive-Bayes e del Perceptron presenti nella libreria Python Scikit-Learn.

2 Struttura del codice

Ciascun file ricrea il corpus delle recensioni, estraendo i file dal database locale, precedentemente scaricato dal sito [1] fornito dagli autori del paper [1].

Tutte le parole e simboli vengono inseriti all'interno di un vettore chiamato corpus. Viene poi creato un vettore labels, lungo quanto il corpus, fatto di 0 e 1. Viene assegnato un uno a tutte le posizioni che nel vettore corpus contengono una parola di una recensione positiva.

Successivamente, usando la libreria Scikit-Learn [2] viene realizzata una matrice Bag-of-Words, dove le colonne rappresentano le singole parole e simboli presenti nelle recensioni, mentre le righe rappresentano le recensioni. Se la parola X è presente nel documento Y, in BoW[X, Y] verrà inserito un uno. Nel caso di calcolo della frequenza delle parole (vedi sotto), ogni volta che una parola viene ripetuta, quel numero incrementa. Resta uno, invece, se viene considerata solo la sua presenza.

In conclusione, il corpus di 1400 recensioni viene diviso in tre parti per permettere la cross-validation.

In dettaglio, ogni file è implementato come segue:

- **unigramsfreq:** per unigram si intende la parola singola o il simbolo di punteggiatura. Per alleggerire il calcolo, vengono scelti solamente gli unigrammi che compaiono più di quattro volte. Questo è l'unico caso in cui si tiene conto della frequenza degli unigrammi.
- **unigramspres:** si tiene conto soltanto della presenza degli unigrammi.
- **unibi:** unigrammi e bigrammi. Vengono considerate anche le coppie di parole, ma per alleggerire l'esecuzione vengono selezionati soltanto quei bigrammi che compaiono più di sette volte.
- **bigrams:** sfruttando la maggioranza del codice in unibi, si selezionano gli indici dei soli bigrammi.
- **unigramsPOS:** grazie alla libreria nltk [3] (in Pang et al usano Qtag di Oliver Mason) si assegna a ciascun unigramma un tag corrispondente al suo ruolo nell'analisi grammaticale.
- **adjectives:** sfruttando i tag della libreria nltk, il corpus viene filtrato per contenere soltanto gli aggettivi.
- **topunigrams:** si limita il corpus agli N unigrammi più usati, dove N è il numero totale di aggettivi presenti in tutto il corpus.
- **position:** gli unigrammi vengono taggati in base alla posizione all'interno della recensione.

Riferimenti bibliografici

[1] Thumbs up? Sentiment Classification using Machine Learning Techniques (Pang et al., EMNLP 2002)

[1] https://www.cs.cornell.edu/people/pabo/movie-review-data/
[2] https://scikit-learn.org/stable/index.html
[3] https://www.nltk.org/

3 Risultati

Caso	# Feature	Naive Bayes	Perceptron
Unigrams freq	12923	79.7%	79.1%
Unigrams	12923	81.2%	79.3%
Unigrams + Bigrams	27699	81.6%	80.3%
Bigrams	14776	78.5%	74.0%
Unigrams + POS	12937	81.4%	80.2%
Adjectives	3228	77.8%	75.4%
Top unigrams	3228	79.6%	78.2%
Position	19960	79.1%	79.1%

Tabella 1: Confronto tra Naive Bayes e Perceptron

We discovered a slight bug in end-of-file handling in our original Naive Bayes code that affected (sometimes negatively, sometimes positively) the first decimal place (i.e., tenths of a percent) of the results reported in our EMNLP 2002 paper. Here are the results:

	corrected	*in paper*		
Features	NB*	NB	ME	SVM
unigrams (freq.)	79.0	78.7	n/a	72.8
unigrams	81.5	81.0	80.4	82.9
unigrams+bigrams	80.5	80.6	80.8	82.7
bigrams	77.3	77.3	77.4	77.1
unigrams+POS	81.5	81.5	80.4	81.9
adjectives	76.8	77.0	77.7	75.1
top 2633 unigrams	80.2	80.3	81.0	81.4
unigrams+position	80.8	81.0	80.1	81.6

Figura 1: Pang's README

La figura 1 si trova sul sito insieme ai database che compongono il corpus. Gli autori vogliono sottolineare l'aver trovato un bug che modifica l'accuratezza dell'implementazione di Naive-Bayes. Sfrutto tale figura per riportare i risultati del paper e, allo stesso tempo, sottolineare come a distanza di poco tempo, gli autori stessi hanno dovuto modificare l'algoritmo e, di conseguenza, i risultati. Con il passare del tempo, dal 2002, anche le implementazioni degli altri algoritmi sono cambiati sensibilmente e da ciò si possono derivare le differenze che si notano tra i risultati nel paper (anche corretti) e questa implementazione. Ciononostante, si possono trarre le stesse conclusioni:

- L'uso della frequenza degli unigrammi risulta controproducente.
- I bigrammi, da soli, non ottengono risultati migliori degli unigrammi.
- Quando gli unigrammi vengono categorizzati in base al loro ruolo grammaticale si ottengono risultati migliori rispetto al solo uso degli unigrammi.
- Come sottolineato in [1], la concezione che gli aggettivi possano indicare meglio il sentimento del testo risulta fallace. Usare lo stesso numero di features, ma scegliendo gli unigrammi più comuni, si rivela essere più efficace.
- In generale, l'uso del Perceptron ottiene risultati inferiori a Naive Bayes.

Oltre a queste osservazioni, è opportuno notare anche che:

- Nell'implementazione moderna l'uso combinato di unigrammi e bigrammi sembra dare risultati migliori.
- Il numero di features è diverso per ogni caso, rispetto al paper. Ho preferito mantenere la logica dietro la scelta (unigrammi che compaiono almeno quattro volte, bigrammi almeno sette volte, confronto degli aggettivi, tagging) invece di forzare un numero apparentemente pre-scelto.