Entropy

Movie Review Summarization Using Supervised Learning and Graph-Based Ranking Algorithm

Team Members :

Amisha Bansal	**-**	**2020201058**
Devesh Jha	**-**	**2020201017**
Pratik Nashine	**-**	**2020201021**

Problem Statement

- Thousand of reviews are posted on daily basis and manual summarization of it is a cumbersome task.
- Summarizing thousands of reviews received by a movie can help the viewer (customer) viewers deciding whether to watch a movie and movie service provider to understand the watching patterns or the interests of their customers.
- An automatic approach is desirable to summarize the lengthy movie reviews so that users can quickly recognize the positive and negative aspects of a movie.

Dataset Description

1. **Pang and Lee polarity dataset** of 2000 movie reviews (version 2). It consists of 1000 positive and 1000 negative reviews.

2. For NB classifier on sentence-level subjectivity classification task **Pang and Lee**, which contains 5000 subjective and 5000 objective sentences taken from movie review summaries and movie plot summaries, respectively.

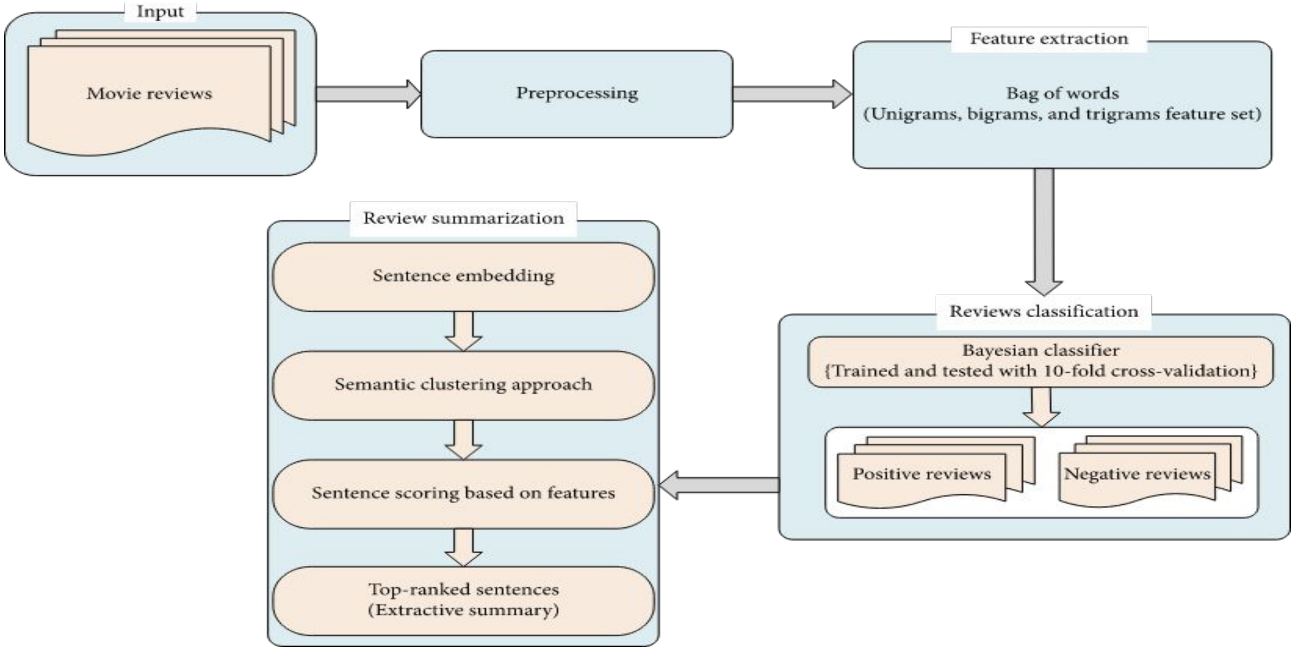
Dataset Description

3. **Andrew**

It consists of 50,000 reviews from the IMDB dataset, and each movie is restricted to have no more than 30 reviews. It is comprised of movie reviews with their corresponding labels (sentiment polarity).Divided into 2.5k training and 2.5k train sets we assume high polarized reviews. The negative reviews are scored ≤4 out of 10, while the positive are scored ≥7 out of 10.

Theory & Model Definition

Overall Flow of Model



Terminology

Sentence : “Reading blogs about data science.”

Unigram : A 1-gram (or unigram) is a one-word sequence. For the above sentence, the unigrams would simply be

“reading”, “blogs”, “about”, “data”, “science”

Bigram : A 2-gram (or bigram) is a two-word sequence of words, like “reading blogs”, “about data”.

TF-IDF(*Term frequency-inverse document frequency***) :** This is a technique to quantify a word in documents, we generally compute a weight to each word which signifies the importance of the word in the document and corpus.

1. Preprocessing of Data and Feature Extraction

- In preprocessing of data , sentence segmentation,tokenization,word stemming is done through NLTK library of python.

- For feature extraction following variations are tried:-
  - Unigrams
  - Bigrams
  - Unigrams + bigrams
  - Unigram frequency + smoothed IDF with cosine normalization
  - Bigram frequency + smoothed IDF with cosine normalization
  - Unigrams + bigrams + smoothed IDF with cosine normalization

2. Review Classification

- In this step users’ reviews are classified using :
  - Multinomial Naive bayes classifier
  - Linear SVM
  - Decision tree
  - CNN.

- Reviews are classified into two classes Positive or Negative.

Multinomial Naive Bayes

Multinomial Naïve Bayes consider a feature vector where a given term represents the number of times it appears or very often i.e. frequency.

In order to classify a new review document, the probability of each term (unigram, bigram, and trigram) in the document's given class label (+ve) is determined, and then the probability of review document's given class label (+ve) is calculated by multiplying the probabilities of all terms with the probability of target class (+ve). Similarly, the probability of review document's given class label (−ve) is calculated.

The review document is classified as positive if its probability of given target class (+ve) is maximized; otherwise, it is classified as negative.

The probability of each term given class c_i , $P(w_k \mid c_i)$, is computed as follows:

$$P\left(w_k \mid \text{positive}\right) = \frac{n_k + 1}{n + |\text{VOC}|},$$

The same process is repeated for negative review document.
Based on the following equation, the review document is assigned to a class if the probability value of the review document’s given class is maximized.

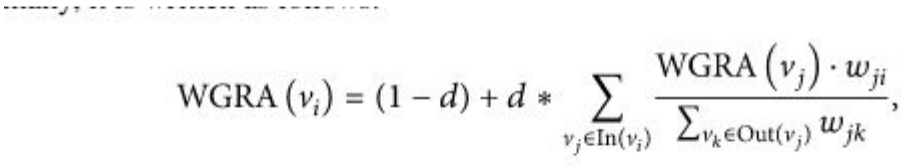
$$C_{NB} = \text{argmax}_{C_i \in C} P\left(C_i\right) \prod_{w \in \text{words}} P\left(\frac{w}{C_i}\right).$$

Review Summarization:

This phase comprises three steps:

- Creation of graph from classified reviews,
- Ranking of graph nodes(review sentences)
- selection of top 20 ranked sentences(nodes) for summary generation.

Ranking of each node is calculated as follows:



[illegible]

$$\mathrm{WGRA}\left(v_i\right) = (1-d) + d * \sum_{v_j \in \mathrm{In}(v_i)} \frac{\mathrm{WGRA}\left(v_j\right) \cdot w_{ji}}{\sum_{v_k \in \mathrm{Out}(v_j)} w_{jk}},$$

d : damping factor

In(V_i) : Number of vertices that are pointing to given vertex V_i

Out(V_j) : Number of outgoing links from the vertex V_j

w_{ji} : represents the weight associated with the edge between nodes V_i and V_j

w_{jk} : represents the weight associated with the outgoing links from vertex V_j

Experiments & Results

Naive Bayes Classification

Features	PL04 Data	IMDB Data	Subjectivity
Unigram	79.0	82.272	86.6
Bigram	74.4	74.432	62.866
Unigram + Bigram	80.6	82.656	86.533
Unigram + Tf-Idf	78.8	82.604	86.833
Bigram + Tf-Idf	77.0	74.416	62.566
Unigram + Bigram + Tf-Idf	78.8	83.008	86.666

Decision Tree Classification

Criteria	PL04 Data	IMDB Data	Subjectivity
Gini	65.2	70.060	75.766
Entropy	60.2	69.892	77.366

SVM & CNN Classification

Classifier	PL04 Data	IMDB Data	Subjectivity
SVM	78.600	83.852	83.433
CNN	49.33	56.468	54.033

Summarization PL04

Scorer	Precision		Recall		F Score	
	Lexrank	Textrank	Lexrank	Textrank	Lexrank	Textrank
Rogue-1	72.42	81.65	73.47	68.02	72.65	73.90
Rogue-2	60.07	70.68	60.96	59.40	60.28	64.30

Summarization IMDB

Scorer	Precision		Recall		F Score	
	Lexrank	Textrank	Lexrank	Textrank	Lexrank	Textrank
Rogue-1	95.41	96.97	85.27	84.88	89.40	89.96
Rogue-2	91.64	93.45	81.64	81.68	85.69	86.60

Individual contributions

1. **Preprocessing and Feature Extraction:** Amisha Bansal and Devesh Jha
2. **Review Classification:** Devesh Jha and Pratik Nashine
3. **Review Summarization:** Pratik Nashine and Amisha Bansal
4. **Results and Reports:** Devesh Jha, Pratik Nashine, Amisha Bansal

Thank You