IST664

NATURAL LANGUAGE PROCESSING

FALL 2023

HOMEWORK 2 – SENTIMENT POLARITY

HENDI KUSHTA

Problem Statement:

The task at hand is to develop a sentiment polarity classifier capable of categorizing a given sentence as either positive or negative. This classifier will be constructed using a "bag-of-words" approach, in which the most frequent words from the training corpus will be selected as word features. The objective is to employ at least a Naive Bayes classifier and implement multi-fold cross-validation to evaluate the classifier's performance. Key evaluation metrics include precision, recall, and F-measure scores.

The problem involves conducting experiments to explore various feature sets and their impact on the classification accuracy. The baseline feature set consists of unigram word features. The goal is to assess whether the introduced features enhance classification accuracy beyond this baseline. Potential experiments include the following:

Stop Word Filtering and Pre-processing: Investigating the impact of filtering out stop words and applying other pre-processing methods to improve feature selection.
Representing Negation: Exploring methods to represent negations within sentences to better capture the nuances of sentiment.
Sentiment Lexicon Integration: Incorporating sentiment lexicons with scores or counts, such as subjectivity measures, to augment the feature set.
Additional Linguistic Features: Considering the inclusion of linguistic features such as adjective words, verbs, noun phrases, and other syntactic or semantic elements to enrich the feature set.
To conduct these experiments, a training dataset needs to be selected. One possible choice is the "sentence_polarity" corpus introduced in class, which provides a well-structured foundation for sentiment analysis. Alternatively, datasets from external sources, such as the sentiment140 dataset, Twitter airline sentiment dataset, or the datasets provided by JHU, can also be considered. The choice of dataset should be informed by factors such as data size, diversity, and relevance to the research goals.

The ultimate objective is to build an accurate and robust sentiment classifier that can effectively categorize sentences as positive or negative, with the flexibility to incorporate various feature engineering techniques to enhance its performance.

Introduction

The foundation of this assignment rests on the "sentence_polarity" corpus, chosen strategically for its expediency and suitability for binary sentiment classification. In a computational context, the dataset's efficiency shines as it demands less processing time compared to other options, making it an ideal choice for resource-sensitive environments like Google Colab, where RAM constraints can be a concern.

What sets the "sentence_polarity" corpus apart is its precision in binary sentiment classification. Each sentence in the corpus is thoughtfully tagged as either 'positive' or 'negative,' simplifying the task for

sentiment analysis models. Binary classification forms the crux of real-world applications where distinguishing between positive and negative sentiment is crucial, such as gauging public opinion on diverse subjects and products.

The journey in this assignment unfolds by systematically organizing the "sentence_polarity" corpus. Each sentence and its individual words are structured to create a comprehensive dataset of documents, with each document representing a single sentence. This dataset is the bedrock upon which sentiment analysis models are trained, acquiring the expertise to discriminate between positive and negative sentiment effectively.

However, this assignment transcends theoretical exercises; it extends into practical applications. Once our models are honed to perfection, they undergo a rigorous evaluation using real-world datasets. These datasets, comprising articles from "true.csv" and "false.csv," serve as the litmus test for our sentiment analysis models. Their capability to accurately capture sentiment within these articles is scrutinized, shedding light on the models' efficacy in practical scenarios. These insights hold significance in grasping the real-world relevance of sentiment analysis across diverse contexts, including news and media analysis.

Pre-processing

Before inputting the training dataset into machine learning models, preprocessing is essential. This preprocessing mainly involves two key steps: tokenization and the removal of non-alphabetic words, such as punctuations. There are various approaches to this process, as it depends on whether capitalization and punctuations convey meaningful sentiment information.

In our simplified model, we've chosen to remove punctuations from the text while retaining words in their original capitalization. This decision is based on the idea that capitalization can carry sentiment-related information, so we want to preserve it. We intentionally avoid converting all text to lowercase because doing so might result in the loss of valuable sentiment cues. In some cases, we even keep certain punctuation marks within the training dataset, as they could hint at the polarity of a sentence.

The goal is to strike a balance between cleaning the text for machine learning and preserving features that genuinely influence sentiment analysis.

Sentiment Classification – Words as Features

The provided code defines a function document_features that generates features for sentiment classification. These features indicate the presence or absence of specific words from a predefined set in a document. The code then creates feature sets for a collection of documents, labeling each feature as 'V_word' and marking them as True or False based on word presence. Finally, it demonstrates how one of these feature sets looks.

```
({'V_the': False,
  'V_a': True,
  'V_and': False,
  'V_of': True,
  'V_to': False,
  'V_is': False,
  'V_it': False,
  'V_that': False,
  'V_in': False,
  'V_as': False,
  'V_but': True,
  'V_film': False,
  'V_with': True,
  'V_this': False,
  'V_for': False,
  'V_its': False,
  'V_movie': False,
  'V_an': True,
```

Classification Model 1: Most frequent words

In terms of performance benchmarking, the Naive Bayes algorithm serves as the reference model.

Cross-validation, particularly the k-fold technique, is employed to evaluate model performance. This method involves dividing the data into k subsets, where the model is trained on k-1 of these subsets and tested on the remaining one. This process iterates k times, with each subset taking its turn as the test set, enabling a robust assessment of the model's effectiveness.

For all three models developed in this project, the following configurations are maintained:

Classifier Algorithm: Naive Bayes Features: The top 1500 most frequent words, denoted as 'V_....' K-fold cross-validation: A method for rigorously evaluating predictive models k = 3 (This value is chosen to manage computational resources) Each model's evaluation is based on key metrics, including:

Accuracy: Quantifies the proportion of correct predictions. Precision: Evaluates the accuracy of positive predictions. Recall: Measures the ability to capture all positive instances. F1 Score: Strikes a balance between precision and recall, particularly for binary classification tasks.

```
Average accuracy: 49.14%
Average precision: 16.24%
Average recall: 33.33%
Average F1 score: 21.84%
```

After running the code, the obtained average performance metrics are as follows:

The average accuracy, representing the proportion of correct predictions, is approximately 49.14%. This indicates that, on average, the classifier's predictions are accurate for about 49.14% of the data

points.

The average precision, which assesses the accuracy of positive predictions, is around 16.24%. This metric reflects the precision of the classifier in correctly identifying positive sentiment instances.

The average recall, measuring the classifier's ability to capture all positive instances, stands at approximately 33.33%. It signifies the classifier's effectiveness in identifying positive sentiment, although there may be some false negatives.

The average F1 score, a balanced metric that considers both precision and recall, is calculated to be approximately 21.84%. This score reflects the trade-off between making accurate positive predictions and ensuring that all positive instances are identified.

Classification Model 2: Most Frequent words without functional words

In constructing this model, we once more employ the top 1500 most commonly occurring words, but this time, we remove the stop or functional words. Drawing inspiration from the lab example, we make an exception for negation words by retaining them and taking their frequency into account. This choice is guided by the understanding that negation words can offer valuable clues regarding the sentiment expressed within a sentence.

```
Average accuracy: 72.73%
Average precision: 73.11%
Average recall: 71.99%
Average F1 score: 72.54%
```

These results illustrate the overall performance of the sentiment classifier when stopwords have been removed from the text data. Specifically:

The average accuracy, representing the proportion of correct predictions, is approximately 72.73%. This suggests that the classifier's predictions are accurate for approximately 72.73% of the data points.

The average precision, which measures the accuracy of positive predictions, is approximately 73.11%. This metric reflects the classifier's precision in correctly identifying positive sentiment instances.

The average recall, indicating the classifier's ability to capture all positive instances, stands at around 71.99%. This signifies the classifier's effectiveness in identifying positive sentiment while minimizing false negatives.

The average F1 score, a balanced metric considering both precision and recall, is approximately 72.54%. This score reflects the trade-off between making accurate positive predictions and ensuring that all positive instances are identified.

Classification Model 3: Most Frequent words without functional words but with Negation Words

In the construction of this model, we once again utilize the top 1500 most frequently appearing words, excluding the stop or functional words. However, in this iteration, we include negation words, such as 'no' or 'not,' which effectively negate the subsequent words. In line with the example from the lab, we opt to negate the word immediately following a negation word. This approach allows us to incorporate

the influence of negation words on sentiment analysis.

The mechanism is rather straightforward: if a word follows a negation word in the text, its corresponding feature is adjusted to signify a negated word. This technique helps capture the nuanced impact of negation in the document's overall sentiment.

```
Average accuracy: 76.48%
Average precision: 77.04%
Average recall: 75.48%
Average F1 score: 76.25%
```

The average performance of the sentiment classifier that takes into account negation words and removes stopwords, employing features that handle negation in the text data, is as follows:

Average accuracy: Approximately 76.48%. This metric represents the proportion of correct predictions, indicating that the classifier's predictions are accurate for approximately 76.48% of the data points.

Average precision: About 77.04%. Precision measures the accuracy of positive predictions. This suggests that the classifier is precise in correctly identifying positive sentiment instances.

Average recall: Approximately 75.48%. Recall assesses the classifier's ability to capture all positive instances, demonstrating the classifier's effectiveness in identifying positive sentiment while minimizing false negatives.

Average F1 score: Around 76.25%. The F1 score provides a balanced measure considering both precision and recall. It reflects the trade-off between making accurate positive predictions and ensuring that all positive instances are identified.

Classification Model 4: Using a sentiment lexicon with scores or counts

A semantic lexicon model, often referred to as a sentiment lexicon or sentiment analysis lexicon, is a computational resource or tool used in natural language processing and text analysis to assign sentiment or emotional polarity (such as positive, negative, or neutral) to words or phrases within a given text.

```
Average accuracy: 74.46%
Average precision: 74.89%
Average recall: 73.63%
Average F1 score: 74.26%
```

This model has an average accuracy of approximately 74.46%, an average precision of 74.89%, an average recall of 73.63%, and an average F1 score of 74.26%. These metrics indicate the overall performance of the sentiment classification model using the sentiment lexicon.

Average Accuracy: The model correctly predicts the sentiment of the text about 74.46% of the time. This means it is relatively accurate in classifying text as positive or negative based on the sentiment lexicon.

Average Precision: The average precision of 74.89% suggests that when the model predicts a positive sentiment, it is correct about 74.89% of the time. For negative sentiment, it is correct about the same percentage of the time. This metric measures how well the model avoids false positives.

Average Recall: The average recall of 73.63% indicates that the model correctly identifies 73.63% of all actual positive instances. It is also able to correctly identify about 73.63% of actual negative instances. This metric measures how well the model avoids false negatives.

Average F1 Score: The average F1 score of 74.26% is the harmonic mean of precision and recall. It provides a balanced measure of the model's overall performance in classifying positive and negative sentiments.

Comparing and Selecting the Best Model

Model	Average Accuracy	Average Precision	Average Recall	Average F1 Score
Model 1: Bag of Words	49.14%	16.24%	33.33%	21.84%
Model 2: No Stopwords	72.73%	73.11%	71.99%	72.54%
Model 3: With Negation	76.48%	77.04%	75.48%	76.25%
Model 4: Sentiment Lexicon Features	74.46%	74.89%	73.63%	74.26%

In summary, Model 3 (with negation handling) outperforms the other three models with the highest accuracy, precision, and F1 score. The second best model is model 4 which is using sentiment lexicon with scores or counts. Model 2, which removes stopwords, also performs significantly better than Model 1. This comparison highlights the importance of preprocessing, including removing stopwords and handling negation, for improving sentiment classification performance.

Prediction for True and Fake News

Perform sentiment analysis on news articles, specifically focusing on categorizing individual sentences within the articles as either having positive or negative sentiment.

The sentiment analysis results for the "True" and "Fake" datasets are summarized below. We analyzed the sentiment of each sentence within the articles and then aggregated the sentiment results for each article.

True News Results:

0	Judge Jeanine Pirro has continued her scream...	6	11
1	A backlash ensued after Donald Trump launched ...	3	9
2	A centerpiece of Donald Trump s campaign, and ...	5	7
3	A new animatronic figure in the Hall of Presid...	8	11
4	Abigail Disney is an heiress with brass ovarie...	12	18
5	Alabama is a notoriously deep red state. It s ...	7	4
6	All Senator John McCain wanted to achieve on t...	4	15
7	Arizona Republican Senator Jeff Flake has neve...	4	8
8	As a Democrat won a Senate seat in deep-red Al...	7	7
9	By now, everyone knows that disgraced National...	4	6
10	By now, the whole world knows that Alabama Sen...	6	11
11	Donald J. Trump spent a portion of his Sunday,...	1	28
12	Donald Trump has a white supremacy problem, an...	8	7
13	Donald Trump held a rally for Alabama Senate c...	1	6
14	Donald Trump is afraid of strong, powerful wom...	4	14
15	Donald Trump just couldn t wish all Americans ...	9	19
16	Donald Trump just signed the GOP tax scam into...	5	9
17	Donald Trump really should have taken his staf...	5	14
18	Donald Trump s current deputy national securit...	3	4
19	Donald Trump s disgraced National Security Adv...	7	5

Fake News Results:

	text	sentiment_pos	sentiment_neg
0	(In Dec. 25 story, in second paragraph, corre...	6	7
1	KING OF PRUSSIA, Pennsylvania/WASHINGTON (Reu...	28	21
2	(Reuters) - A U.S. appeals court in Washington...	6	8
3	(Reuters) - A U.S. appeals court on Friday sai...	10	8
4	(Reuters) - A gift-wrapped package addressed t...	3	6
5	(Reuters) - A lottery drawing to settle a tied...	8	6
6	(Reuters) - Alabama officials on Thursday cert...	7	13
7	(Reuters) - Democrat Doug Jones' surprise vict...	4	5
8	(Reuters) - The U.S. Congress on Thursday appr...	6	14
9	(Reuters) - The U.S. State Department has told...	17	12
10	LIMA (Reuters) - Peru's President Pedro Pablo ...	14	16
11	MEXICO CITY (Reuters) - Mexico's finance minis...	2	1
12	NEW YORK (Reuters) - A federal appeals court i...	12	11
13	NEW YORK (Reuters) - A federal judge in New Yo...	11	7
14	NEW YORK (Reuters) - The U.S. Justice Departme...	12	7
15	NEW YORK/WASHINGTON (Reuters) - The new U.S. t...	13	10
16	SAN FRANCISCO (Reuters) - A second U.S. judge ...	7	5
17	SEATTLE/WASHINGTON (Reuters) - President Donal...	16	24
18	The following statements were posted to the ve...	3	9
19	The following statements were posted to the ve...	8	8

Sentiment Analysis Findings

To examine whether fake content tends to contain more positive or negative sentences compared to true content, several statistical analyses were conducted. The following results and tests help in understanding the sentiment distribution in fake and true news articles:

Mann-Whitney U Test for Positive Sentence Ratio:

A Mann-Whitney U test was performed to compare the positive sentence ratio between true and fake articles.
Result: True and fake articles significantly differ in positive sentence ratio, suggesting that the distribution of positive sentences varies between the two categories.
Mann-Whitney U Test for Negative Sentence Ratio:

Another Mann-Whitney U test was conducted, this time for the negative sentence ratio.
Result: Similar to the positive sentence ratio, true and fake articles significantly differ in negative sentence ratio, indicating variations in the distribution of negative sentences between the two categories.
Average Negative Sentences:

The average count of negative sentences in both fake and true news articles was calculated.
Result: Fake news articles have an average of approximately 9.76 negative sentences, while true news

articles have an average of around 8.5 negative sentences. This suggests that fake news tends to contain a slightly higher number of negative sentences on average.
Average Positive Sentences:

The average count of positive sentences in fake and true news articles was also computed.
Result: Fake news articles have an average of about 6.02 positive sentences, whereas true news articles have an average of roughly 7.26 positive sentences. True news contains a higher average of positive sentences.
Average Negative Sentence Ratio:

The average negative sentence ratio, which is the proportion of negative sentences relative to all sentences, was determined for both fake and true news.
Result: Fake news articles have an average negative sentence ratio of approximately 0.6207, while true news articles have an average ratio of approximately 0.5535. This suggests that fake news contains a slightly higher proportion of negative sentences compared to true news.
Average Positive Sentence Ratio:

Similarly, the average positive sentence ratio, representing the proportion of positive sentences to all sentences, was calculated for both categories.
Result: Fake news articles have an average positive sentence ratio of around 0.3793, whereas true news articles have an average ratio of about 0.4465. This indicates that true news contains a higher proportion of positive sentences.
In summary, the statistical tests and calculations reveal that there are notable differences in the distribution of positive and negative sentences and their respective ratios between fake and true news articles. Fake news tends to have a slightly higher average of negative sentences and a lower proportion of positive sentences compared to true news.

Discussion – What I learned through this assignment

Through this sentiment analysis assignment, I have gained valuable insights into the methodologies and processes involved in building and evaluating sentiment classification models. Here are the key takeaways:

Choice of Dataset: Selecting the right dataset is critical. The "sentence_polarity" corpus, chosen for its efficiency and suitability for binary sentiment classification, proved to be an excellent choice. It simplifies the task of distinguishing between positive and negative sentiment, making it an ideal foundation for training models.
Data Preprocessing: Preprocessing, including tokenization and the removal of non-alphabetic words, is essential. Striking a balance between cleaning the text for machine learning and preserving features that genuinely influence sentiment analysis is crucial. In our case, retaining capitalization for sentiment cues and retaining some punctuation marks for sentiment hints was the chosen approach.
Feature Selection: The choice of features significantly impacts classification accuracy. Experimenting with different feature sets, including stop word filtering, negation representation, sentiment lexicons, and additional linguistic features, demonstrates that feature engineering can enhance the classifier's performance.
Classifier Choice: We employed the Naive Bayes classifier as the reference model, considering its performance as the baseline. Evaluating models using k-fold cross-validation (in our case, k=3) provides a robust assessment of their effectiveness.
Model Comparison: After running the models, we compared their performance. Model 3, which

incorporates negation handling and removes stopwords, outperformed the other three models in terms of accuracy, precision, and F1 score. he second best model is model 4 which is using sentiment lexicon with scores or counts. Model 2, with stopwords removed, also demonstrated significant improvements compared to Model 1.

Real-world Application: The assignment extends beyond theory into practical applications. It involves sentiment analysis on real-world datasets, specifically categorizing individual sentences within news articles as positive or negative. This application demonstrates the relevance of sentiment analysis in diverse contexts, including news and media analysis.

Sentiment Analysis Findings: The sentiment analysis on "True" and "Fake" news articles revealed differences in sentiment distribution. Fake news tends to contain slightly more negative sentences and a lower proportion of positive sentences compared to true news. These findings have implications for understanding the sentiment characteristics of news articles in different categories.

In summary, this assignment has provided a comprehensive understanding of sentiment analysis, from dataset selection and preprocessing to feature engineering, model evaluation, and real-world applications. It underscores the importance of feature selection and preprocessing in improving classification accuracy and highlights the practical relevance of sentiment analysis in media and news analysis.