TLN Part 3 Lab 05 - Topic Modeling and Text Visualization

Paletto Andrea, Tuninetti André

September 12, 2023

Contents

1 Introduction **3**

2 Methodology **3**
2.1 Function Overview 3
2.1.1 Topic Modeling with LDA 3

3 Conclusion **4**
3.1 Evaluation 4

1 Introduction

In this document, we present an overview of the implementation of topic modeling and text visualization. The task was:

- Topic Modeling
  - Extracting Topics from a Corpus
  - Consider using the Gensim library for topic modeling.
    https://radimrehurek.com/gensim/
- Text Visualization: Selection of Text Data Visualization Libraries (Any of your choice)
  - Explore various libraries for visualizing textual data.

2 Methodology

2.1 Function Overview

We have implemented several functions to achieve our goals:

- `get_data`: This function retrieves text summaries from Wikipedia based on a list of specified topics using the Wikipedia API.
- `processing_data`: This function processes the retrieved text data by tokenizing it into sentences, removing common words (stop words) and punctuation, and returning processed and original documents.
- `generate_synsets`: This function retrieves synsets (sets of related words) from WordNet for a given list of topics and stores them in a data structure. It also includes related hyponyms and hypernyms for each synset.
- `obtain_synset`: This function attempts to infer a topic from a list of synsets. It initializes variables to track the maximum value and a counter. Then, it iterates through the provided list of synsets, counting the occurrences of each synset. The function identifies the synset with the maximum occurrence and prints it as the inferred topic for each set of synsets in the list.

2.1.1 Topic Modeling with LDA

This section covers the steps for topic modeling using Latent Dirichlet Allocation (LDA):

1. Process the text data using the `processing_data` function.
2. Create a Gensim Dictionary to assign IDs to tokens.
3. Create a Gensim corpus to represent the tokenized text data.
4. Perform LDA topic modeling with different numbers of topics and compute coherence scores.
5. Select the best number of topics based on coherence scores.
6. Train the LDA model with the chosen number of topics.
7. Display the topics using the `show_topics` function of the LDA model.
8. Prepare the data for visualization using the `prepare` function from pyLDAvis.
9. Display the prepared data for interactive visualization.
10. Extract keywords, explore semantic relationships, and infer topics.
11. Calculate perplexity to evaluate model performance.

3 Conclusion

In this exercise, we employed a training set consisting of three main topics: Nutrition, Technology, and Sport. We gathered information from Wikipedia using specific keywords for each topic:

- For Nutrition: 'Food', 'Junk Food', 'Vegan'
- For Technology: 'Artificial Intelligence', 'Internet of Things', 'Robotics'
- For Sport: 'Baseball', 'Football', 'Basketball'

3.1 Evaluation

For evaluating the performance of our models we use Coherence and perplexity, there are two metrics used to evaluate the quality of topic modeling results. In our case, we obtained the following scores for different numbers of topics:

Coherence Scores Coherence measures how interpretable or meaningful the topics generated by your topic modeling algorithm are. Higher coherence values generally indicate that the topics make more sense and are more distinct from each other. The best coherence score is achieved when using 3 topics, with a coherence value of approximately 0.546. This suggests that the topics extracted when using 3 topics are relatively coherent and interpretable.

Perplexity Scores Perplexity is a measure of how well a language model predicts a sample of text. Lower perplexity values indicate that the model is better at predicting the data. In your case, the best perplexity score is achieved when using 5 topics, with a perplexity value of approximately -12.04. This suggests that the model with 5 topics is better at predicting the text data compared to the models with fewer topics.

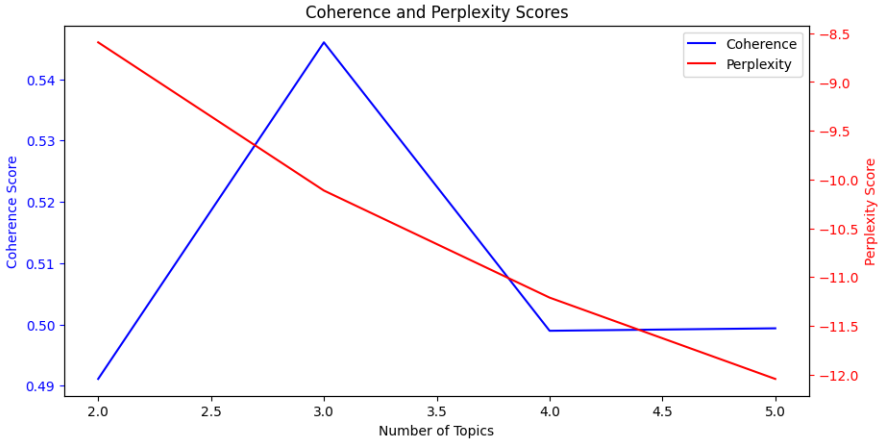

Figure 1: plot with coherence and perplexity values

The reason why the best coherence score doesn't match the best perplexity score is that these two metrics are measuring different aspects of our topic modeling results:

- Coherence is concerned with the quality and interpretability of the topics themselves.
- Perplexity is concerned with how well the model, with its given set of topics, predicts the data it was trained on.

It's common for the optimal number of topics based on coherence and perplexity to differ because these metrics have different goals. Sometimes, you might need to strike a balance between having interpretable topics (higher coherence) and a model that predicts the data well (lower perplexity).

We achieved a coherence score of 0.52 when using all three topics simultaneously, indicating that the generated text was more logically connected and coherent.

However, when it comes to perplexity, we observed a score of -10.25. This negative perplexity score can be attributed to Gensim's automatic conversion of very small probabilities to the log scale. Although we typically aim for lower perplexity scores, it is crucial to understand that a lower bound value suggests a decline in the model's performance. This means that, as the perplexity value decreases, the quality of the generated text by the model may actually deteriorate.