**Final Reprot of Document Comparison and Topic Modeling Tool**

**Motivation**

In the realm of text analysis and information retrieval, there exists a pressing need for tools that can delve deeper into the content of documents, beyond mere surface-level similarities. Our proposed software tool aims to address this gap by offering a comprehensive solution for comparing multiple documents and extracting common topics and frequently occurring words through advanced natural language processing techniques. The envisioned software tool will be a standalone application, primarily intended for researchers, students, content creators, and professionals engaged in analyzing and comparing textual content. It is also designed for ordinary people to easily compare documents. The tool's major functions will include document comparison to identify similarities and differences, topic modeling using a mixture language model with Probabilistic Latent Semantic Analysis (PLSA) to cluster words related to common topics, semantic analysis to understand the context and meaning of these words, and a user-friendly visualization of the results.

**Design**

We adopted the client-server design pattern and have observed several benefits. We developed our application using a combination of Django and React. Django is a Python-based server framework that handles the core functionalities of the application, while React is a frontend framework used primarily for crafting the web user interfaces. This design enhances scalability, as each component can be scaled independently according to demand. It also improves maintainability, as separating the frontend and backend allows developers to focus on specific areas without interference. Furthermore, this separation facilitates easier updates and faster deployment cycles.

**Model**

PLSA:

PLSA (Probabilistic Latent Semantic Analysis) is a statistical technique for text data, primarily used to discover latent topics within large collections of documents. Introduced by Thomas Hofmann in 1999, PLSA posits that each document is a mixture of latent topics, where each topic is characterized by a distribution over words. This model establishes probabilistic links between documents and words, revealing the underlying semantic structure of the document collection.

The PLSA model assumes that documents are generated by first choosing a topic for each word from a distribution of topics specific to the document, and then choosing the word from a distribution specific to the chosen topic. The parameters of PLSA are typically estimated using the Expectation-Maximization (EM) algorithm. The E-step computes the probability of each topic given each word, while the M-step updates the distribution of topics and the distribution of words under each topic.

LDA

Latent Dirichlet Allocation (LDA) is a widely used topic modeling technique that automatically discovers latent topics within large document collections. Introduced by Blei, Ng, and Jordan in 2003, LDA builds on the foundation of PLSA but incorporates a Bayesian probability framework, enhancing the model's generalizability and stability.

The core idea of LDA is that documents are generated from a mixture of topics, where each topic is characterized by a probability distribution over words. The model presupposes that certain topics (such as economics, politics, education) appear frequently across specific collections of documents, with each topic closely associated with specific vocabulary.

In LDA, the distribution of topics within a document and the distribution of words within a topic are both governed by Dirichlet distributions. A document's words are generated by first choosing a topic from the document's topic distribution and then selecting a word from that topic's word distribution. The model's parameters are typically estimated using variational Bayesian methods or Gibbs sampling, which help infer the parameters of the probabilistic model from real document collections.

**Implement**

Post process:

Before the text is fed into the model, we need to perform some additional processing. First, we convert all words in the text to lowercase. Then, we use the nltk's stopword library to remove all words from the text that do not have significant meaning. After that, we apply the WordNetLemmatizer method to transform the words into their root forms, which can enhance the computational speed of the model and the accuracy of the results.

Core function:

For LDA, we opted for the LDA model from gensim. We tuned the parameters to generate five topics, and then selected the top 20 high-probability words for each topic in each document. On the frontend, we took the intersection of high-probability words from two documents to identify potential common themes between them.

For PLSA, we used the PLSA model implemented by laserwave[1]. Similarly, we obtained the five topics with the highest probabilities and their 20 words. However, for selecting common themes, we chose the topic with the highest probability generated from a corpus composed of the two documents.

**Similarity measurement**

We have chosen cosine similarity, BERT similarity, and Pearson correlation to reflect the similarity between two documents, allowing for a more comprehensive assessment of their resemblance.

Cosine similarity:

Cosine similarity is a metric used to measure how similar two vectors are irrespective of their size. Commonly used in natural language processing and information retrieval, cosine similarity calculates the cosine of the angle between two vectors projected in a multi-dimensional space. This approach can effectively determine how similar two documents (or any text items) are likely to be in terms of their content.

BERT similarity:

BERT similarity refers to the application of BERT (Bidirectional Encoder Representations from Transformers) models to compute semantic similarity between pieces of text. Developed by Google, BERT revolutionized the field of natural language processing by using a deep learning approach based on the Transformer architecture to generate rich, context-aware embeddings for text.

Pearson correlation:

Pearson correlation coefficient, also known as Pearson's r, is a statistical measure that expresses the degree of linear relationship between two variables. It is widely used in the fields of statistics, psychology, social science, biology, and economics among others, to quantify the degree to which two variables are related.

**Limitations and Future Work**

Currently, our application can only accept two documents as input and does not yet support the comparison of multiple documents. We may implement a feature for comparing multiple documents in the future. Additionally, our application lacks the functionality to save and record results, so the final outcomes cannot be preserved. We hope to develop a feature to record results in the future.

**References**

[1] [laserwave/plsa: a python implementation of probabilistic latent semantic analysis (plsa) using EM algorithm (github.com)](https://github.com/laserwave/plsa)