From Vectors to Transformers: Decoding Search Engine Performance

Akshat Meena and Siddharth Singh

Department of Computer Science and Engineering
Indian Institute of Technology Madras

May 14, 2023

Introduction

- Search engines play a crucial role in information retrieval by finding relevant and meaningful results from vast amounts of data.
- There are many techniques/models used to build a search engine. We have used the Vector Space Model (VSM), Latent Semantic Indexing (LSI) and a variant of Bidirectional Encoder Representations from Transformers (BERT) as part of the project.
- Techniques like VSM and LSI involve weights for different terms. Various weighting schemes are be used:
  - Basic Count
  - TF-IDF
  - Normalized TF-IDF
  - Glasgow Model
- Each approach has its own benefits and limitations, and we intend to do a comparative study between these approaches/techniques.

Vector Space Model

- Vector Space Model is a basic but popular search engine model. It represents documents and queries as vectors in a high-dimensional space. It then uses similarity calculations between vectors to find the documents relevant to a given query.
- Each term in the corpus is a dimension in the space where the documents and the queries are projected. The coefficient of each dimension represents the weight or importance of the term in that particular document or query.
- Various weighting schemes can be employed, such as basic counts, term frequency-inverse document frequency (TF-IDF), normalized term frequency-inverse document frequency, and the Glasgow Model.

Contd

Advantages

- VSM is a simple model to implement compared to the other modern methods.
- It gives sufficiently good results when the query is "similar" to the documents.
- The performance of the VSM model can be enhanced by using sophisticated weighting schemes.

Limitations

- It doesn't consider the order in which the words come together to form a sentence. It relies on the "bag-of-words" representation of the queries and documents, due to which it is not able to extract the semantic meaning of the text accurately.
- It is very computation-intensive as the space in which the documents and queries are embedded is generally very high-dimensional.
- All the vectors have to be recomputed once a new term is added to the term space.

Latent Semantic Indexing

- Latent Semantic Indexing extracts "hidden" concepts from the corpus that relate the terms to the documents in which they appear using matrix factorization.
- A term-document matrix is constructed where all the terms present in the corpus form the rows of the matrix, and all the documents form the columns of the matrix.
- The entries of the matrix signify the relevance of that term to the document. Different weighting schemes can be used here.
- Singular Value Decomposition (SVD) is used to decompose the matrix into three matrices - a term-concept matrix, a singular value matrix, and the concept-document matrix.
- The number of concepts to be extracted is a hyperparameter, and its value is to be decided by the user.

Contd

Advantages

- It does not require any source of external knowledge.
- It is good at handling synonymy, which means it can capture the relation between two different words that convey the same meaning.
- The vector space in which the queries and documents are mapped is of lower dimensionality than VSM. Hence, constructing the vectors and calculating the cosine similarities is less computation-intensive.

Limitations

- Lower number of hidden dimensions gives better recall, worse precision, whereas a higher number of hidden dimensions gives worse recall, better precision.
- LSI is inefficient at handling polysemy.
- Performing the SVD is a computationally intensive task.

Weighting Techniques

1. Basic Count

$$w_i = tf_i$$

2. TF-IDF

$$w_i = tf_i * log\left(\frac{D}{df_i}\right)$$

3. Normalized TF-IDF

$$w_{ij} = \frac{tf_{ij}}{max\ tf_{ij}} * log\left(\frac{D}{df_i}\right)$$

and the weight of term i in query Q can be written as:

$$w_{Qi} = \left(0.5 + 0.5 * \frac{tf_{Qi}}{max\ tf_{Qi}}\right) * log\left(\frac{D}{df_i}\right)$$

4. Glasgow Model

$$w_i = \frac{log(tf_{ij} + 1)}{log(length_j)} * log\left(\frac{D}{df_i}\right)$$

Transformers

- The transformer architecture is based on the concept of self-attention mechanisms.
- They capture global dependencies and contextual information efficiently by attending to all positions or words within a sequence simultaneously.
- Input sequences are represented as embeddings, which are then transformed by multiple self-attention layers and feed-forward neural networks.
- The self-attention mechanism allows each word to attend to all other words in the sequence, capturing their contextual importance.
- This enables the model to learn intricate relationships between words and improve its understanding of the semantic meaning and structure of the text.

Contd

Advantages

- They can handle long-range dependencies and capture contextual information effectively.
- Transformers have beaten the state-of-the-art models in various NLP tasks, such as machine translation, text summarization, and question-answering.

Limitations

- It requires substantial computational resources for training due to its large number of parameters.
- Fine-tuning transformer models on domain-specific data can be time-consuming and may necessitate a significant amount of annotated data.
- Transformers are sensitive to the quality and diversity of the training data, which can affect their generalization capabilities.

Experimental Methodolgy

- We analyzed the VSM and LSI models with different weighting schemes, namely basic counts, term frequency-inverse document frequency (TF-IDF), normalized term frequency-inverse document frequency (TF-IDF) and the Glasgow model.
- We experimented with the LSI model with different hyperparameter values, which in this case was the number of hidden dimensions.
- In both models, preprocessing was done on the queries and the documents. Segmentation, tokenisation, stemming, and stopword removal were performed in order. Then, the preprocessed queries and documents were passed to the models for information retrieval.
- We ran the multi-qa-MiniLM-L6-cos-v1 model on the documents and queries to find their encodings. After this, cosine similarity was used as a similarity measure to rank the documents in order of their relevance to a given query.

Contd

- However, we did not preprocess the queries or the documents in the case of the multi-qa-MiniLM-L6-cos-v1 model.
- We calculated different evaluation metrics like precision, recall, f-score, Mean Average Precision (MAP), and normalized Discounted Cumulative Gain (nDCG) for each model at different ranks.
- We compared the values of these evaluation metrics for different models and weighting schemes and made some observations based on them.
- We finally performed statistical hypothesis testing to verify them.

Hypothesis Testing

- We used the Wilcoxon signed-rank test for our purpose as it doesn't assume the data samples to be normally distributed, which was the case with our data.
- The Wilcoxon signed-rank test is a non-parametric statistical test used to compare two related or paired samples.
- It is an alternative to the paired t-test when the assumptions of the t-test, such as the normality of the differences or interval-level data, are not met.
- The test is particularly useful when dealing with ordinal or non-normally distributed data.
- If we have a specific expectation or hypothesis about the direction of the difference, we can use a directional alternative hypothesis.
- The directional alternative hypothesis can be one-sided, either specifying that the median difference is greater than zero (one-tailed upper) or that the median difference is less than zero (one-tailed lower).

References

- NLP course slides.
- http://www.minerazzi.com/tutorials/term-vector-3.pdf
- https://www.analyticsvidhya.com/blog/2019/06/understanding-transformers-nlp-state-of-the-art-models/
- https://www.analyticsvidhya.com/blog/2019/09/demystifying-bert-groundbreaking-nlp-framework/
- https://machinelearningmastery.com/hypothesis-test-for-comparing-machine-learning-algorithms
- https://machinelearningmastery.com/statistical-hypothesis-tests-in-python-cheat-sheet/