Semantic Similarity Detecting

Jingyi Zhang jyz0328@bu.edu
Ruoxi Jin jrx99@bu.edu

Github for this project files: https://github.com/jyz0328/cs505-project

Abstract

The goal of this miletsont is to evaluate different models for **Semantic Similarity Detection**. We assess the performance of traditional models such as **Bag-of-Words (BoW)** and **TF-IDF** in measuring **Semantic Textual Similarity (STS)**. For this, we use the **STS Benchmark (STSb)** dataset, which consists of sentence pairs annotated with human-generated similarity scores, normalized between 0 and 1. Our primary evaluation metric is **Cosine Similarity**, with **Pearson Correlation** and **Spearman's Rank Correlation** used to measure how well the predicted similarity scores align with the actual scores.

The **BoW** and **TF-IDF** models serve as baselines. While they efficiently capture lexical overlap between sentences, they struggle to grasp deeper semantic relationships. Our evaluation shows that **BoW** tends to underpredict similarity when the true similarity is high, while **TF-IDF** provides slightly better predictions by emphasizing important words. However, both models perform moderately in **Pearson** and **Spearman** correlations, highlighting their limitations in capturing semantic meaning.

In future work, we plan to evaluate transformer-based models such as **Sentence-BERT (SBERT)**, which generate semantic embeddings and better capture contextual meaning, demonstrating significant improvements over traditional methods in the task of **Semantic Similarity Detection**.

Data source explanation

We will be using the STSb dataset（https://huggingface.co/datasets/sentence-transformers/stsb）for our Semantic Textual Similarity task. This dataset is split into three parts: `stsb_test.csv`, `stsb_train.csv`, and `stsb_validation.csv`. Each dataset includes pairs of sentences (Sentence1 and Sentence2) and their corresponding similarity score.

The **Semantic Textual Similarity Benchmark** (Cer et al., 2017) is a collection of sentence pairs derived from various sources like news headlines, video captions, and natural language inference data. Each pair is annotated with a human-generated similarity score, ranging from 1 to 5. In this specific version of the dataset, the similarity scores have been normalized to a range between 0 and 1. Therefore, no additional data preprocessing is needed.

Below is an example of the first ten entries from the training dataset (`stsb_train.csv`):

sentence1	sentence2	score
A plane is taking off.	An air plane is taking off.	1
A man is playing a large flute.	A man is playing a flute.	0.76
A man is spreading shreded cheese on a pizza.	A man is spreading shredded cheese on an uncooked pizza.	0.76
Three men are playing chess.	Two men are playing chess.	0.52
A man is playing the cello.	A man seated is playing the cello.	0.85
Some men are fighting.	Two men are fighting.	0.85
A man is smoking.	A man is skating.	0.1
The man is playing the piano.	The man is playing the guitar.	0.32
A man is playing on a guitar and singing.	A woman is playing an acoustic guitar and singing.	0.44
A person is throwing a cat on to the ceiling.	A person throws a cat on the ceiling.	1

Evaluation metrics

In our project, **Cosine Similarity** will be the primary evaluation metric. Since we are using the SentenceTransformer model to generate sentence embeddings, Cosine Similarity is the best metric for measuring the similarity between two vectors. The value of Cosine Similarity ranges from -1 to 1, where 1 indicates complete similarity, 0 indicates no correlation, and -1 indicates complete dissimilarity.

Explanation: Cosine Similarity is used to compute the angular similarity between two vectors and is particularly well-suited for measuring the similarity between text embeddings.

Formula:

Cosine Similarity=A⋅B ∥A∥ ∥B∥ \text{Cosine Similarity} = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|}Cosine Similarity=∥A∥ ∥B∥A⋅B

where A\mathbf{A}A and B\mathbf{B}B are the vector representations of two texts.

Use case: When using models like BERT or SentenceTransformer to convert text into vector representations, Cosine Similarity is an effective metric to measure the similarity between sentences.

Additionally, to evaluate the alignment between the model's predicted similarity scores and the actual similarity scores provided in the dataset, we will also use **Pearson Correlation** and **Spearman's Rank Correlation**. These metrics help assess the strength and direction of the relationship between the predicted and actual similarity scores.

Pearson Correlation: Pearson Correlation measures the strength of the linear relationship between two variables, with values ranging from -1 to 1. A value of 1 indicates a perfect positive linear correlation, -1 indicates a perfect negative linear correlation, and 0 indicates no linear relationship. It is calculated by normalizing the covariance of the two variables. The formula is as follows:

Pearson Correlation=∑(Xi−X‾)(Yi−Y‾)∑(Xi−X‾)2∑(Yi−Y‾)2\text{Pearson Correlation} = \frac{\sum (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum (X_i - \bar{X})^2 \sum (Y_i - \bar{Y})^2}}Pearson Correlation=∑(Xi−X‾)2∑(Yi−Y‾)2∑(Xi−X‾)(Yi−Y‾)

where XiX_iXi and YiY_iYi are the values of the two variables, and X‾\bar{X}X‾ and Y‾\bar{Y}Y‾ are their respective means.

Use case: Pearson Correlation is used to measure the linear relationship between predicted similarity scores and the actual scores, making it suitable for cases where a linear dependency is expected.

Spearman's Rank Correlation: Spearman’s Rank Correlation is a non-parametric measure that assesses how well the relationship between two variables can be described using a monotonic function. It ranks the data and calculates the correlation between the ranks. If the rankings of the two variables are perfectly aligned, the Spearman coefficient is 1; if the rankings are completely opposite, it is -1. The formula is:

Spearman Correlation=1−6∑di2n(n2−1)\text{Spearman Correlation} = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)}Spearman Correlation=1−n(n2−1)6∑di2

where did_idi is the difference between the ranks of corresponding values of two variables, and nnn is the number of data points.

Use case: Spearman’s Rank Correlation is useful when the relationship between the variables is monotonic but not necessarily linear, making it robust for datasets with outliers or when linear assumptions do not hold.

Baseline Methods

We will use a simple **Bag-of-Words (BoW)** model as a baseline method. In this approach, each sentence is represented as a vector of word frequencies, and the similarity between two sentences is calculated using **Cosine Similarity** between these vectors.

Bag-of-Words (BoW) Model:
The BoW model creates a vocabulary from all words in the dataset and represents each sentence as a vector, where each element corresponds to the frequency of a word in the sentence. This approach provides a quick way to compare the surface-level word overlap between sentences, but it lacks the ability to capture deeper semantic relationships. For example, two sentences with different word choices but similar meanings may still be considered dissimilar by BoW.

- **Formula** for Cosine Similarity:

Cosine Similarity=A⋅B // A // // B // \text{Cosine Similarity} = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|}Cosine Similarity= // A // // B // A⋅B

where A\mathbf{A}A and B\mathbf{B}B are the word frequency vectors of two sentences.

Advantages: BoW is simple and computationally efficient, making it useful for quickly assessing word overlap.
Limitations: BoW cannot capture the meaning or order of words, thus failing to reflect the true semantic similarity between sentences.

Another baseline we will implement is **TF-IDF (Term Frequency-Inverse Document Frequency)**, which improves upon BoW by considering not only the frequency of words in a sentence but also the importance of a word within the entire corpus.

TF-IDF Model:
The **TF-IDF** approach refines BoW by assigning more weight to words that are frequent in a specific sentence but rare across the corpus, making them more important for distinguishing sentences. Each word in the sentence is assigned a TF-IDF score, calculated as the product of its term frequency (TF) and its inverse document frequency (IDF).

- **Term Frequency (TF)**:

TF=number of occurrences of the word in the sentencetotal number of words in the sentence\text{TF} = \frac{\text{number of occurrences of the word in the sentence}}{\text{total number of words in the sentence}}TF=total number of words in the sentencenumber of occurrences of the word in the sentence

- **Inverse Document Frequency (IDF)**:

IDF=log(total number of documentsnumber of documents containing the word+1)\text{IDF} = \log\left(\frac{\text{total number of documents}}{\text{number of documents containing the word} + 1}\right)IDF=log(number of documents containing the word+1total number of documents)

- **TF-IDF score**:

TF-IDF=TF×IDF\text{TF-IDF} = \text{TF} \times \text{IDF}TF-IDF=TF×IDF

TF-IDF also uses **Cosine Similarity** to compare sentence vectors, where each vector represents the TF-IDF weights of the words in the sentence.

Advantages: TF-IDF improves upon BoW by highlighting important words and downplaying commonly used ones, thus offering better sentence comparison.
Limitations: Like BoW, TF-IDF still cannot capture the deeper semantic relationships between words and sentences, such as word meaning or sentence context.

These baselines (BoW and TF-IDF) are easy to implement and provide a foundation for comparison against more sophisticated approaches like **Sentence-BERT (SBERT)**, which generates sentence embeddings that capture richer semantic relationships.

Baseline performance Evaluation

In this section, we evaluate the performance of the **Bag-of-Words (BoW)** and **TF-IDF** models using the same metrics—**Cosine Similarity**, **Pearson Correlation**, and **Spearman's Rank Correlation**—discussed earlier. The purpose of this evaluation is to establish a baseline for these traditional models and provide a point of comparison for more advanced approaches like **Sentence-BERT (SBERT)**.

To facilitate this evaluation, we implemented a Python script called **`sum.py`**, which can be found in our github() which performs the following tasks:

1. **Loading the Data**: The script reads in sentence pairs and similarity scores from `stsb_train.csv`.
2. **Feature Extraction**: It applies both the **BoW** and **TF-IDF** models to each sentence pair to generate feature vectors.

3. **Similarity Calculation**: Using these feature vectors, the script calculates the **Cosine Similarity** between each pair of sentences for both BoW and TF-IDF models.
4. **Performance Evaluation**: The script then computes the **Pearson Correlation** and **Spearman Rank Correlation** between the predicted Cosine Similarity scores and the actual similarity scores from the dataset.
5. **Data Processing**: Finally, it outputs the processed data, including the original sentences, actual similarity scores, and the predicted similarity scores from both BoW and TF-IDF models.

Here is the output of running `sum.py` along with a brief explanation of the results. The script was executed using the `stsb_train.csv` dataset.

```
zhangjingyi@zhangs-MacBook-Pro milestone2 % python3 sum.py

BoW Cosine Similarity Scores: [0.8164965809277261, 0.8944271909999159,
0.7559289460184544, 0.7999999999999999, 0.912870929175277]

BoW Pearson Correlation: 0.6155996890240858

BoW Spearman Rank Correlation: 0.5954448588531247

TF-IDF Cosine Similarity Scores: [0.8335532357732083,
0.8338802875083728, 0.6116755444885693, 0.8733179643127706,
0.8020562036162421]

TF-IDF Pearson Correlation: 0.7040039456980351

TF-IDF Spearman Rank Correlation: 0.6800503056821073
```

The output shows the following:

- **BoW Cosine Similarity Scores**: The predicted similarity scores for five sample sentence pairs using the BoW model.
- **BoW Pearson and Spearman Correlations**: The correlation between the BoW model's predicted scores and the actual similarity scores.
- **TF-IDF Cosine Similarity Scores**: The predicted similarity scores for the same sentence pairs using the TF-IDF model.
- **TF-IDF Pearson and Spearman Correlations**: The correlation between the TF-IDF model's predicted scores and the actual similarity scores.

Additionally, here are the first 10 lines of the processed data from the script, including the original sentences, actual similarity scores, and the predicted scores from both BoW and TF-IDF models:

sentence1	sentence2	score	BoW score	TF-IDF score
A plane is taking off.	An air plane is taking off.	1	0.816497	0.833553
A man is playing a large flute.	A man is playing a flute.	0.76	0.894427	0.833880
A man is spreading shreded cheese on a pizza.	A man is spreading shredded cheese on an uncooked pizza.	0.76	0.755929	0.611676
Three men are playing chess.	Two men are playing chess.	0.52	0.800000	0.873318
A man is playing the cello.	A man seated is playing the cello.	0.85	0.912871	0.802056
Some men are fighting.	Two men are fighting.	0.85	0.750000	0.803670
A man is smoking.	A man is skating.	0.10	0.666667	0.206987
The man is playing the piano.	The man is playing the guitar.	0.32	0.875000	0.614865
A man is playing on a guitar and singing.	A woman is playing an acoustic guitar and singing.	0.44	0.668153	0.639648

A person is throwing a cat on to the ceiling.	A person throws a cat on the ceiling.	1	0.721688	0.605611

This table shows a side-by-side comparison of the actual similarity scores with the predicted scores from both models, providing insight into how well each model captures sentence similarity.

Here are more detailed explanations about the performance result.

1. Bag-of-Words (BoW) Results

The **BoW** model, while simple and computationally efficient, struggled to accurately capture semantic relationships between sentences. This is expected, as BoW only considers lexical overlap without any understanding of context or word meaning. The performance metrics reflect these limitations:

Cosine Similarity Scores: For a few sample sentence pairs, the BoW model generated the following Cosine Similarity scores:

```
[0.816, 0.894, 0.756, 0.800, 0.913]
```

- **Pearson Correlation**: The correlation between BoW's predicted Cosine Similarity scores and the actual similarity scores was **0.6156**, indicating a moderate positive linear relationship.
- **Spearman Rank Correlation**: The Spearman correlation was **0.5954**, which shows a moderate alignment in the rank-ordering of sentence pairs based on similarity.

Overall, BoW demonstrated a modest ability to identify lexical similarity, but its inability to handle semantic nuances led to suboptimal performance, especially for sentence pairs that share little lexical overlap but have similar meanings.

2. TF-IDF Results

The **TF-IDF** model improves upon BoW by considering the importance of words within the broader dataset, which allows it to perform slightly better in identifying sentence similarity. However, like BoW, it still cannot capture sentence context or deeper semantic relationships.

Cosine Similarity Scores: For a few sample sentence pairs, TF-IDF produced the following Cosine Similarity scores:

```
[0.834, 0.834, 0.612, 0.873, 0.802]
```

-
- **Pearson Correlation**: TF-IDF achieved a Pearson correlation of **0.7040**, showing a stronger linear relationship between predicted and actual similarity scores compared to BoW.
- **Spearman Rank Correlation**: TF-IDF’s Spearman correlation was **0.6800**, indicating better rank-order consistency than BoW.

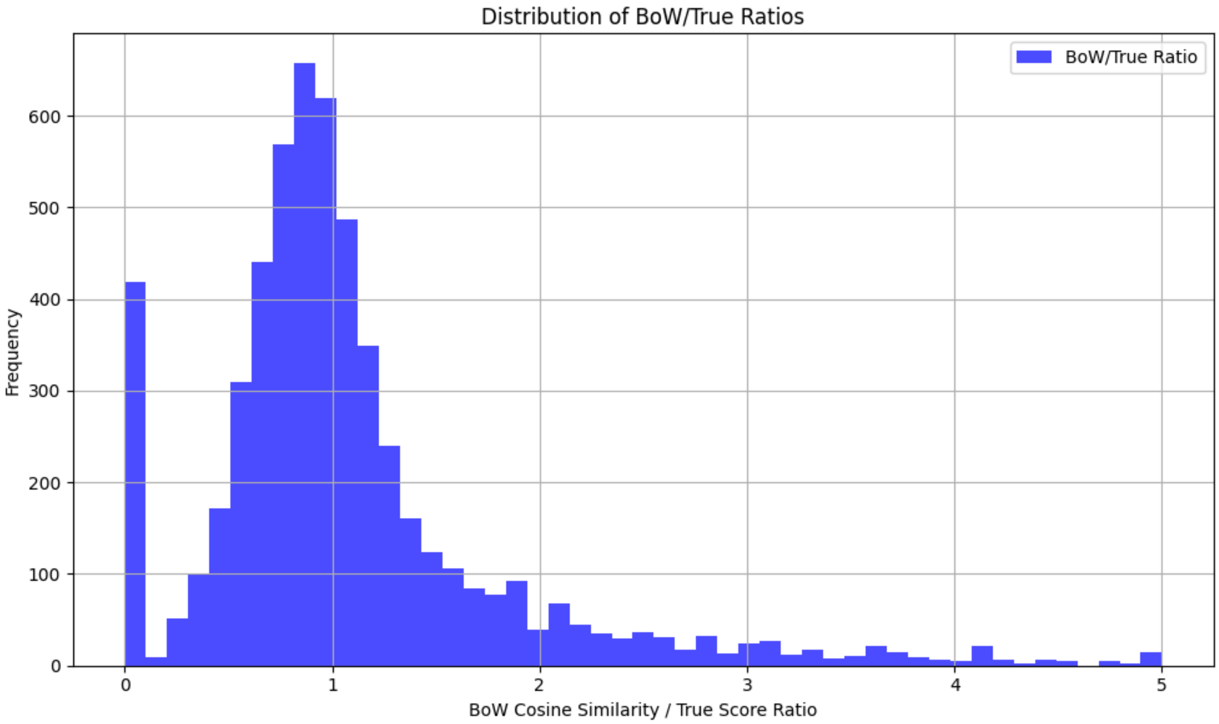
While TF-IDF outperformed BoW, its limitations are still evident, particularly in its inability to capture the true meaning of sentences beyond lexical word weighting.

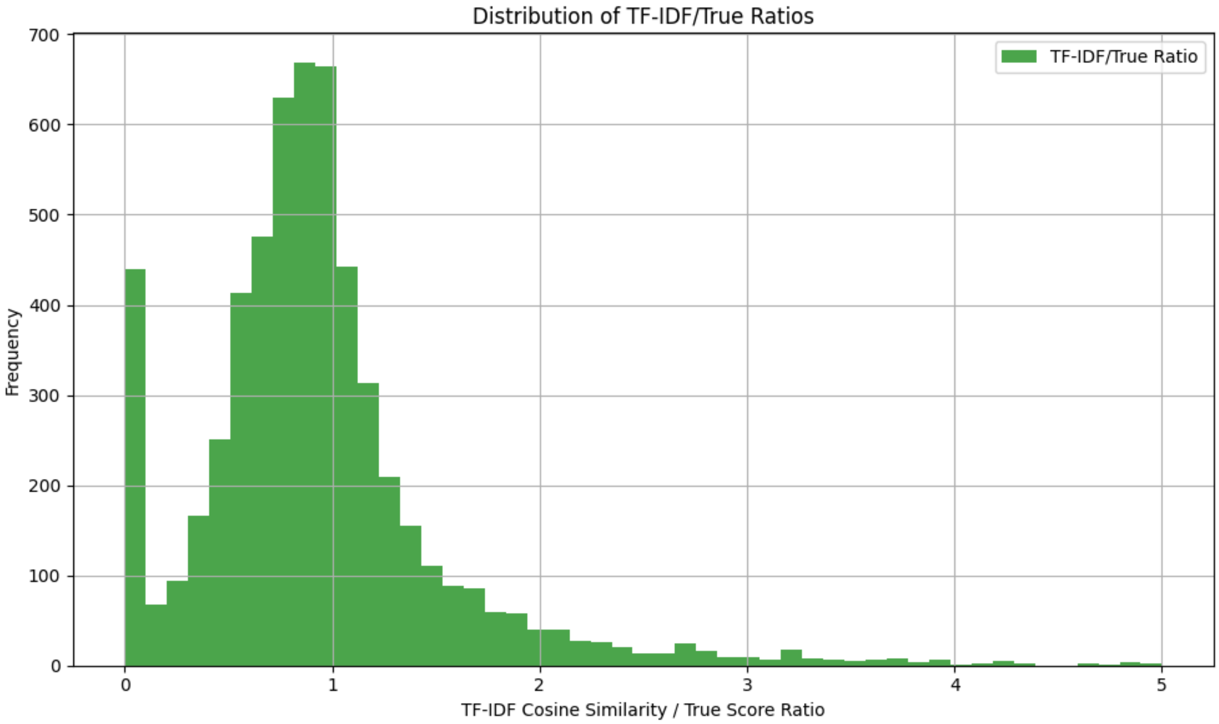
3. Analysis of BoW/True and TF-IDF/True Ratios

To further explore the behavior of these baseline models, we calculated the ratios of predicted similarity scores to the actual similarity scores, denoted as **BoW/True** and **TF-IDF/True**. These ratios allow us to assess how well the models' predictions align with the true similarity scores.

In the figures below, we show the **BoW/True** and **TF-IDF/True** ratios against their **frequency**. **Frequency** refers to how many times a particular ratio appears. For example, if **BoW/True** = 0.9 appears 400 times, then the **frequency** for that ratio is 400. This visualization helps to understand the distribution of prediction errors and how often different levels of over- or under-prediction occur.

- **BoW/True Ratio**: The majority of the BoW/True ratios were concentrated between 0 and 1, indicating that BoW tends to underpredict similarity when the true similarity is high. This is consistent with the model’s reliance on word overlap, which penalizes sentences with different lexical choices but similar meanings.
- **TF-IDF/True Ratio**: The TF-IDF/True ratio distribution showed more consistency around 1, suggesting that TF-IDF generally predicted similarity scores closer to the true values. However, there were still some instances where TF-IDF either overestimated or underestimated the similarity.





4. Discussion of Results

The results from both **BoW** and **TF-IDF** highlight the limitations of these traditional models:

- **BoW** is primarily useful for identifying lexical overlap but fails to capture any semantic meaning.
- **TF-IDF** improves by assigning more weight to important words but still lacks the ability to understand the contextual or semantic relationships between sentences.

These results establish a baseline for comparison with more sophisticated models, such as **Sentence-BERT (SBERT)**. While **TF-IDF** offers marginal improvements over **BoW**, neither model can truly handle the complexities of sentence meaning, which is critical for tasks like sentence similarity detection.

Next Steps - Transformer-Based Model Evaluation

Next, we will evaluate the performance of **Sentence-BERT (SBERT)**, a transformer-based model that generates semantic embeddings for sentences. By leveraging context and deeper word meaning, SBERT is expected to significantly outperform both BoW and TF-IDF in this task. We will use the same evaluation metrics—Cosine Similarity, Pearson Correlation, and Spearman's Rank Correlation—to directly compare SBERT's performance to the baselines and demonstrate the advantages of using semantic embeddings for sentence similarity tasks.