Homonyms Problem Report

1 Introduction

1.1 The Problem: The Failure of Keyword-Based Sentiment Analysis

In natural language, the same word can carry different meanings and emotional tones depending on the surrounding context. Users of modern applications expect systems to understand this.

Standard techniques, such as those based on keyword matching or "bag-of-words" models, are effective at identifying documents with direct keyword overlap but can't grasp the true semantic intent of a user's query.

This project addresses this challenge, as in the problem statement:

- "I hate the selfishness in you" $\rightarrow$ **Negative Sentiment**
- "I hate any one who can hurt you" $\rightarrow$ **Positive Sentiment**

Here, a simple model would see the word "hate" and incorrectly classify both sentences as negative.

1.2 Project Goal and Accomplishment

The goal of this project is to develop, compare, and evaluate two different pipelines to solve this contextual sentiment problem for a custom-built corpus of nuanced English sentences.

1. **A Baseline Static Embedding Model:** A search engine was built using pre-trained **GloVe** word embeddings with a **Logistic Regression** classifier. This represents a strong classical approach that understands word meanings but lacks contextual awareness.
2. **A Modern Contextual Model Comparison:** A comprehensive evaluation was performed on several state-of-the-art, pre-trained **Transformer models** (includingBERT and RoBERTa).
3. **A Targeted Contextual Probe:** A final experiment was conducted to analyze the models' sensitivity to explicit contextual cues.

2 List of Tools and Technologies Used

This project was developed in a Python-based Google Colab environment, leveraging GPU acceleration for deep learning models. The key technologies and libraries used are categorized below.

2.1 Core Libraries

- **Pandas & NumPy:** Used for data manipulation, preparation, and numerical operations.
- **Scikit-learn:** Utilized for building the baseline model (`LogisticRegression`), performing stratified data splits (`train_test_split`), and evaluating all models (`classification_report`, `accuracy_score`).
- **PyTorch:** Served as the deep learning backend for running the Transformer models.
- **Hugging Face Transformers:** The core library for accessing and deploying pre-trained models. Its high-level `pipeline` function was used for efficient sentiment analysis inference.

2.2 Pre-trained Models & Resources

- **GloVe (glove.6B.200d.txt):** Provided the static word embeddings for the non-contextual baseline model. Sourced from the Stanford NLP Group.
- **Transformer Models (from Hugging Face Hub):**
  - **DistilBERT (distilbert-base-uncased-finetuned-sst-2-english):** A fast and efficient contextual model used for comparative analysis.
  - **RoBERTa (siebert/sentiment-roberta-large-english):** A larger, more powerful contextual model used to test the limits of out-of-the-box performance.

3 Data Description

3.1 Dataset

To address the project's focus on contextual sentiment, a standard, large-scale dataset (such as IMDb or SST-2) was not suitable. Such datasets rarely contain the specific, nuanced examples required for a focused analysis.

Instead, a custom "micro-dataset" was hand-crafted specifically for this project. This dataset was designed to test the core problem. Its features are:

- **Instance Composition:** The dataset consists of **62 sentences** in total. The majority of these are structured in **pairs**, where a single keyword or phrase (e.g.,

"broke the bank," "fired," "cold") is used in two different sentences to convey opposite sentiments.

- **Balanced Labels:** The dataset is perfectly balanced, with an equal number of 'positive' and 'negative' examples. This prevents a classifier from gaining accuracy by simply guessing the majority class.
- **Features:** The dataset contains two primary columns:
  - Sentence: The raw text input (the feature).
  - Label: The ground truth sentiment, either 'positive' or 'negative' (the target). For modeling, this was converted to a numeric format (1 for positive, 0 for negative).

3.2 Division Between Training and Testing

The dataset was divided into two distinct sets: a **Training Set** and a **Test Set**.

3.2.1 A Note on the Absence of a Validation Set

A separate validation set, typically used for hyperparameter tuning, does not exist in this project's design. This decision was based on the specific goals of the experiments:

1. **Baseline Model (GloVe + Logistic Regression):** The objective was to test the conceptual limitation of static embeddings. While the Logistic Regression classifier has tunable hyperparameters, optimizing them was not the focus. Using the default, well-established parameters provides a fair and representative baseline without the need for a validation-driven search.
2. **Advanced Models (Pre-trained Transformers):** The core of the advanced experiment was to evaluate the **out-of-the-box performance** of several powerful, pre-trained models. As these models were used directly for inference without any fine-tuning, there were no hyperparameters to optimize, rendering a validation set unnecessary for this stage.

Therefore, a two-part split (Train/Test) was the most direct and appropriate methodology for this comparative analysis.

3.2.2 Splitting Process

The process was as follows:

1. **Isolation of Core Challenge Sentences:** The two key sentences provided in the assignment prompt ("I hate the selfishness in you" and "I hate any one who can hurt you") were manually separated from the main pool of data. This was a critical step to prevent data leakage and to guarantee their presence in the final test set.

2. **Stratified Splitting:** The remaining **60 sentences** from the custom dataset were then split into a training set and an initial test set using a **75/25 ratio**. A stratify parameter was used to ensure that the proportion of positive and negative labels was identical in both splits, preserving the dataset's balance.

3. **Final Set Creation:**

    - The **Training Set** consists of the **45 sentences** from the 75% split. This data was used to train the baseline GloVe + Logistic Regression model.
    - The **Test Set** was created by combining the **15 sentences** from the 25% split with the **2 manually isolated challenge sentences**.

This methodology resulted in the following final data division:

- **Total Instances:** 62
- **Training Set Size:** 45 instances (72.6%)
- **Testing Set Size:** 17 instances (27.4%)

4 Baseline Experiment: Static Embeddings with GloVe

4.1 Goal

The goal of the baseline experiment was to establish a strong benchmark using a classical, non-contextual NLP approach. A model was built that is powerful enough to understand the meaning of individual words but is architecturally incapable of comprehending how word order and surrounding context alter that meaning.

This model is intentionally designed to expose the exact weaknesses highlighted in the project's problem statement. The chosen architecture for this baseline was a combination of pre-trained **GloVe (Global Vectors for Word Representation)** for word embeddings and a **Logistic Regression** classifier.

4.2 Experimental Steps & Results

The experiment was conducted using the defined training and testing sets.

1. **Feature Extraction:** Pre-trained 200-dimensional GloVe vectors (glove.6B.200d.txt) were loaded. Each sentence in both the training and testing sets was converted into a single, fixed-size vector by averaging the GloVe vectors of its constituent words. This "average meaning" vector served as the numerical input for the classifier.

2. **Model Training:** A Logistic Regression classifier was trained on the sentence vectors derived from the **45 sentences** in the training set. The model learned to associate these averaged vectors with their corresponding positive or negative sentiment labels.

3. **Evaluation:** The trained classifier was then used to make predictions on the unseen **17 sentences** in the test set.

Results: The baseline model performed surprisingly well in terms of overall metrics, achieving a final accuracy of **82.35%**. The detailed performance is summarized below.

Table 1: Classification Report (GloVe Baseline)

	Precision	Recall	F1-Score	Support
Negative	0.80	0.89	0.84	9
Positive	0.86	0.75	0.80	8
Accuracy			0.82	17
Macro Avg	0.83	0.82	0.82	17
Weighted Avg	0.83	0.82	0.82	17

However, a qualitative analysis of the model's performance on the two core challenge sentences reveals the critical limitation of this approach.

Table 2: Analysis of GloVe on Core Challenge Sentences

	Sentence	True Label	Predicted Label	Correct
15	I hate the selfishness in you	negative	negative	True
16	I hate any one who can hurt you	positive	negative	False

4.3 Conclusion of Baseline Experiment

The baseline experiment was successful in achieving its goal. The high overall accuracy of **82.35%** confirms that the GloVe + Classifier approach is a strong and valid benchmark.

However, the analysis of the specific challenge sentences definitively demonstrates the fundamental flaw of static, non-contextual embeddings.

- The model correctly classified "I hate the selfishness in you" as negative, as the strong negative signals from "hate" and "selfishness" dominated the averaged vector.
- Critically, it **failed** on the nuanced sentence, predicting "I hate any one who can hurt you" as **negative**. The model's logic is transparent and flawed: it saw the word "hate," and its prediction was driven by this powerful negative keyword, completely ignoring the protective and positive intent of the surrounding phrase.

This result perfectly illustrates the problem statement. The baseline model, while statistically capable, is brittle and lacks true semantic understanding. This motivates the need for more advanced, context-aware models to solve the problem reliably.

5 Other Experiments: Contextual Embeddings with Transformer Models

Having shown the limitations of the static embedding baseline, the next step of the project is to evaluate modern, context-aware deep learning models. These experiments were designed to determine if state-of-the-art Transformer architectures could overcome the contextual challenges where the GloVe model failed.

5.1 Experiment 1: Comparative Analysis of Pre-trained Sentiment Classifiers

The goal of this experiment was to evaluate and compare the "out-of-the-box" performance of several popular, pre-trained Transformer models on our specific, nuanced test set. The objective was to identify the best-performing model and to analyze its ability to correctly classify the core challenge sentences without any further training or fine-tuning.

1. **Candidate Selection:** Two high-performing, pre-trained sentiment analysis models were selected from the Hugging Face Hub, each with a different architecture or training domain:
    - **BERT:** "distilbert-base-uncased-finetuned-sst-2-english".
    - **RoBERTa:** "siebert/sentiment-roberta-large-english".
2. **Evaluation:** Each model was loaded via the Hugging Face pipeline for sentiment analysis. The entire test set of **17 sentences** was passed to each model for inference.
3. **Results:** All Transformer models significantly outperformed the GloVe baseline in terms of overall accuracy and demonstrated a superior understanding of context. The "siebert/sentiment-roberta-large-english" model emerged as the top performer. A summary of the overall accuracy is below:

Table 3: Overall Performance of Transformer Models

Model	Accuracy	Notes
BERT	94.12%	High precision for negative class
RoBERTa (Sentiment)	94.12%	Top performer

Table 4: Classification Report (BERT)

	Precision	Recall	F1-Score	Support
Negative	0.90	1.00	0.95	9
Positive	1.00	0.88	0.93	8
Accuracy			0.94	17
Macro Avg	0.95	0.94	0.94	17
Weighted Avg	0.95	0.94	0.94	17

Table 5: Classification Report (RoBERTa)

	Precision	Recall	F1-Score	Support
Negative	0.90	1.00	0.95	9
Positive	1.00	0.88	0.93	8
Accuracy			0.94	17
Macro Avg	0.95	0.94	0.94	17
Weighted Avg	0.95	0.94	0.94	17

However, the most critical result was the performance on the two challenge sentences.

Table 6: Analysis of Transformer Models on Core Challenge Sentences

	Sentence	True Label	BERT	RoBERTa
15	I hate the selfishness in you	negative	negative	negative
16	I hate any one who can hurt you	positive	negative	negative

Every tested Transformer model correctly classified "I hate the selfishness in you" as negative. Critically, **every model also incorrectly classified "I hate any one who can hurt you" as negative**.

This experiment demonstrates two key findings:

1. Contextual models like BERT and RoBERTa are demonstrably superior to the static GloVe baseline, achieving a significant boost in overall accuracy.

2. Despite their power, all tested pre-trained models share a systematic limitation. Their training on massive, general-purpose datasets has created a powerful statistical bias, causing them to associate the word "hate" with negative sentiment so strongly that they fail to grasp the nuanced, protective intent of the key challenge sentence.

5.2 Experiment 2: Probing Model Bias with Contextual Anchors

The goal of this final experiment was to test the hypothesis from the previous experiment: that the models' failure was due to a powerful but potentially overridable statistical bias. This experiment was designed to probe the models' sensitivity to context by introducing explicit positive "anchor" phrases to the challenge sentence.

1. **Sentence Augmentation:** A list of augmented sentences was created, adding different types of positive context to the original challenge sentence.
2. **Evaluation:** The same set of pre-loaded Transformer pipelines from the previous experiment was used. Each model was tasked with classifying the full list of original and augmented sentences.
3. **Results:**

Table 7: Impact of Different Contextual Anchors Across Models

Sentence	BERT	RoBERTa (Sentiment)
I hate any one who can hurt you	negative (98.15%)	negative (99.41%)
I love you and I hate any one who can hurt you	positive (99.92%)	positive (99.75%)
you are my friend I hate any one who can hurt you	positive (99.84%)	positive (99.53%)
Here is my opinion: I hate any one who can hurt you	negative (93.77%)	negative (99.23%)

Table 8: Analysis of GloVe Performance on Challenge & Augmented Sentences

Sentence	GloVe Prediction
I hate any one who can hurt you	negative
I love you and I hate any one who can hurt you	positive
you are my friend I hate any one who can hurt you	positive
Here is my opinion: I hate any one who can hurt you	positive

6 Overall Conclusion

The Semantic Engine, powered by pre-trained Transformer models, is the best approach. The top-performing model, `siebert/sentiment-roberta-large-english`, achieved an impressive **94.12% accuracy**, a significant improvement over the 82.35% accuracy of the GloVe baseline.

However, the most critical finding of this project is the identification of a systematic limitation shared by all tested "out-of-the-box" Transformer models. **Every advanced model failed on the key challenge sentence, "I hate any one who can hurt you,"** classifying it as negative due to a powerful statistical bias learned from their general-purpose training data.

The final experiment, which involved augmenting the sentence with positive contextual anchors, proved that this bias is not immutable. The models' ability to unanimously flip their predictions to positive when presented with phrases like "I love you..." demonstrates their latent capacity for nuanced understanding.

7 How to re-produce the results

Run in colab environment the homonyms_problem.ipynb notebook from the repo.

8 Questions

8.1 What was the biggest challenge you faced when carrying out this project?

The biggest challenge was dealing with the subtle ambiguity of homonyms in sentiment analysis. For example, a sentence like *"I hate anyone who can hurt you"* is inherently protective and positive in intent, but most models, including advanced contextual ones like BERT and RoBERTa, misclassified it as negative. This highlighted a key limitation: small datasets and surface-level lexical cues are insufficient for capturing pragmatic meaning.

8.2 What do you think you have learned from the project?

From this project, I learned three lessons:

1. **The importance of baselines:** Even a simple GloVe + Logistic Regression pipeline can provide valuable insights when compared against more advanced contextual models.

2. **The limitations of pretrained models without fine-tuning:** While BERT-based models generally outperform static embeddings, they can still fail in nuanced cases without additional context or domain-specific fine-tuning.

3. **The role of context in sentiment analysis:** By experimenting with contextual augmentation (e.g., adding relational or emotional anchors), I saw how providing richer context can drastically improve classification.