NLP - Home Assignment 1

Ameer Ahmed 324993690
Mohamed Jabali 212788293
Ibrahim Abomokh 315270678

November 27, 2024

Section 2: Optimizing word2vec

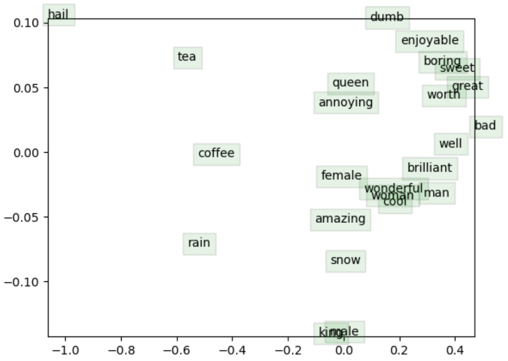

Figure 1: Word Vector Plot

Analysis and Interpretation:
The plot visualizes word vectors in 2D space, where the proximity between words suggests their semantic similarity.
Clusters/Trends:

- Words like *enjoyable*, *sweet*, and *boring* form a cluster, suggesting they are related to **subjective experiences** or **opinions**.
- The words *king*, *queen*, *man*, and *woman* group together, forming a cluster associated with **gender** and **social roles**.
- *Rain* and *snow* appear near each other, which makes sense as they are both **weather-related terms**. However, *hail* being close to *rain* is unexpected. Hail, being ice-based, should be more distinct from rain.

- Words like *amazing*, *brilliant*, and *wonderful* are close, indicating they share a **positive sentiment**.

Quality of Word Vectors: The vectors capture broad relationships well, but some words, like *hail* and *snow*, appear too close. This suggests that the word embeddings might need fine-tuning or the training data didn't sufficiently differentiate certain terms. While relationships like *king* to *queen* are accurately captured, finer distinctions might require more sophisticated models.
Improvements: To improve the quality of word embeddings, consider the following:

- Train the model on a more specific or larger dataset, particularly one focused on weather-related terms for better distinction between *hail* and *snow*.
- Fine-tune the embeddings using **contextual embeddings** (e.g., BERT, RoBERTa) to capture richer semantic relationships.
- Adjust hyperparameters such as increasing the embedding dimensions or extending the training duration for better differentiation between words.

Section 3: Optimizing word2vec

3.a Finding the distribution that maximizes the objective

Given θ that maximizes the objective function, we can think of the problem as a constrained optimization problem. Since we are dealing with probabilities, each θ must satisfy:

$$(*) \quad \sum_{o' \in V} p_\theta(o' \mid c) = 1 \quad \text{for every } c \in V$$

In addition, for each $o, c \in V$, they appear in the objective exactly $\#(c, o)$ times. Hence, we can rewrite the objective as:

$$J(\theta) = \sum_{c \in V} \sum_{o \in V} \#(c, o) \cdot p_\theta(o \mid c)$$

Using the constraints mentioned above, we can apply Lagrange multipliers to obtain:

$$L = \sum_{c \in V} \sum_{o \in V} \#(c, o) \cdot p_\theta(o \mid c) + \sum_{c \in V} \lambda_c \left(1 - \sum_{o \in V} p_\theta(o \mid c) \right)$$

We know that the critical points of J and L are the same. Now, we take the derivative of L with respect to $p_\theta(o \mid c)$:

$$\frac{\partial L}{\partial p_\theta(o \mid c)} = \frac{\#(c, o)}{p_\theta(o \mid c)} - \lambda_c$$

Setting the derivative to zero, we get:

$$\frac{\#(c,o)}{p_\theta(o \mid c)} = \lambda_c \qquad \Longrightarrow \qquad (**) \quad p_\theta(o \mid c) = \frac{\#(c,o)}{\lambda_c}$$

From $(*)$ and $(**)$, we obtain:

$$\sum_{o' \in V} p_\theta(o' \mid c) = 1 \qquad \Longrightarrow \qquad \sum_{o' \in V} \frac{\#(c,o')}{\lambda_c} = 1 \qquad \Longrightarrow \qquad \lambda_c = \sum_{o' \in V} \#(c,o')$$

Substituting λ_c back into $(**)$:

$$p_\theta(o \mid c) = \frac{\#(c,o)}{\sum_{o' \in V} \#(c,o')}$$

3.b Example for an imposable optimal solution

Any feasible solution satisfies that for every $c, o \in V$:

$$p(o \mid c) = \frac{\exp(u_o \cdot v_c)}{\sum_{o' \in V} \exp(u_{o'} \cdot v_c)} > 0$$

Consider the following vocabulary $V = \{a, b, c\}$ and the following corpus $C = \{aa, aa, ab\}$. From the previous section, we obtain that the optimal solution must satisfy:

$$p(a \mid a) = \frac{\#(a,a)}{\sum_{o \in V} \#(a,o)} = \frac{2}{3}$$

$$p(b \mid a) = \frac{\#(a,b)}{\sum_{o \in V} \#(a,o)} = \frac{1}{3}$$

$$p(c \mid a) = \frac{\#(a,c)}{\sum_{o \in V} \#(a,o)} = 0$$

This is impossible to achieve because $p(o \mid c)$ must be strictly greater than 0 for all $o \in V$.

Section 4: Paraphrase Detection

4.a

Note that $0 \leq \text{ReLU}(x_1)^T\text{ReLU}(x_2)$. Additionally, the function σ is strictly increasing and satisfies $\sigma(0) = 0.5$, so for all $x \geq 0$, we know that $\sigma(x) \geq 0.5$. Based on this, we can conclude that:

$$\forall x_1, x_2 : p(\text{the pair is a paraphrase} \mid x_1, x_2) > 0.5$$

This means we will always predict the pair as a paraphrase. Given that the ratio of positive to negative examples is 1:2, the accuracy is simply:

$$\text{Accuracy} = \frac{1}{1+2} = \frac{1}{3}.$$

4.b

Alternatively, instead of using the ReLU function, we could directly use the dot product. This would define the probability as:

$$p(\text{the pair is a paraphrase} \mid x_1, x_2) = \sigma(x_1^T x_2).$$

This would allow for the output to potentially be negative, which means we could predict non-paraphrases as well. However, the downside is that by removing the ReLU, we lose the non-linear transformation it provides, which could limit the model's ability to capture more complex patterns in the data.

4.c

- **Accuracy**: This is the percentage of correctly predicted instances (both true positives and true negatives) out of the total number of instances.

  However, in imbalanced datasets, accuracy can be misleading. For example, if the positive class is rare and the model simply predicts the majority class, it could achieve a high accuracy without actually learning to detect the minority class (e.g., paraphrases).

- **Precision**: This metric tells us how many of the detected paraphrases are actually correct (true positives out of all predicted positives).

  Precision is especially important when false positives are costly. For instance, in a paraphrase detection task, we want to minimize the number of incorrect predictions, so precision helps us understand how reliable the model is when it says something is a paraphrase.

- **Recall**: This is the percentage of correctly predicted positives (true positives) out of all the actual positives (true positives + false negatives).

  Recall is key when missing positive instances (false negatives) is costly. In tasks like paraphrase detection, you want to make sure the model captures as many paraphrases as possible, even at the cost of some false positives.

- **ROC-AUC (Receiver Operating Characteristic - Area Under the Curve)**: This metric evaluates how well the model distinguishes between the positive and negative classes, taking into account various threshold settings.

  While ROC-AUC is useful in many cases, it may not be the best choice for highly imbalanced datasets. It doesn't focus enough on the performance of the minority class, which is often the class that we care about the most in these scenarios.

- **AUC-PR (Precision-Recall Area Under the Curve)**: This metric gives us a better understanding of the trade-off between precision and recall at different thresholds.

  AUC-PR is particularly valuable for imbalanced datasets because it highlights how well the model is identifying the positive class (the minority class). It is often more informative than ROC-AUC in these situations.

- **Confusion Matrix**: A confusion matrix is a table that shows the breakdown of true positives, true negatives, false positives, and false negatives.

  When evaluating a paraphrase detection model, where detecting the positive class is more important, the PR-AUC metric is usually a better choice. PR-AUC focuses more on the performance of the positive class, works well with imbalanced data, and gives us a fuller view of precision-recall tradeoffs, which are essential for this task.

Section 6: Topic modeling

6.1.a

Coherence: This metric measures the degree to which words in a topic frequently appear together in documents. It reflects how interpretable or meaningful the topic is.
Cluster Evaluation: This metric assesses the clustering quality of the topics, i.e., how well the model can group similar documents together based on shared topics.

6.1.b

- **AG News**: A popular dataset for news articles classified into multiple topics such as: World, Sports, Business and Science/Technology

  It is widely used for text classification tasks where articles are categorized into these predefined topics.

- **20 Newsgroups**: A dataset consisting of newsgroup documents organized into 20 different categories, making it ideal for experiments in text classification and topic modeling. Categories include: Politics, Sports,Technology, Religion,and more.

- **New York Times Annotated Corpus**: A dataset of news articles with metadata such as publication dates, authors, and categories. It is particularly suitable for: Topic modeling, Content analysis and Studying the evolution of media coverage over time.

6.3

Model: BERTopic trained on the 20 Newsgroups dataset.
Evaluation Metrics:

- **u-mass: -9.49**

  suggests that the words in the topic are rarely found together in the same documents.

- **c-v: 0.41**

  indicates moderate coherence,

6.4

Topic Interpretability and Semantics:
While coherence scores capture word co-occurrence and statistical relationships, they don't necessarily reflect whether a human can easily understand and interpret a topic. For example, a topic with high coherence may contain words that

statistically co-occur but still feel abstract or disjointed to a human reader. Humans can assess whether a topic is intuitively meaningful or whether the words truly belong together in a conceptual sense

Diversity of Topics:

Metrics like coherence do not measure topic diversity—how distinct or varied the topics are. Human evaluators can assess whether topics are sufficiently diverse to cover the range of themes in the dataset. Too much overlap between topics might indicate that the model is failing to capture distinct thematic structures