Report on Dense Embedding Model Training

Submitted By: Malaika Ahmed (492471)
Submission Date: November 14, 2024

---

1. Approach to Model Selection, Training, and Evaluation

Model Selection:

- Chosen Model: Word2Vec
- Reasoning:

> Captures Word Relationships: Word2Vec is effective in understanding word relationships based on context.

> Suitable for Smaller Datasets: Given the dataset size (10 pages with 20-25 lines each), Word2Vec is appropriate.

> Flexibility: Allows easy adjustment of embedding dimensions and is quick to train.

Hyperparameters:

- Vector Size: 30 dimensions
- Context Window Size: 5 words
- Minimum Count: 1 (to include all words)
- Training Algorithm: Skip-gram (sg=1)
- Negative Sampling: 5 samples

Training Process:

- Data Preparation: Text files were extracted, normalized, tokenized, and lemmatized.
- Model Training: The Word2Vec model was trained on the processed text data for 10 epochs using 4 CPU cores.

Model Summary:

- Vocabulary Size: 1030 words
- Embedding Dimension: 30

2. Challenges Encountered and Solutions

Challenges:

Data Size:

> **Issue:** The dataset was relatively small, consisting of only 10 pages with 15-20 lines each. This limited the amount of data available for training the embedding model, which can impact the quality and robustness of the embeddings.

> **Impact:** Smaller datasets can lead to less accurate and less generalizable embeddings because the model has fewer examples to learn from.

Approach to tackle the issue:

Model Choice:

> **Solution:** Opted for Word2Vec, which is known to perform well with smaller datasets. Word2Vec can still provide high-quality embeddings without needing a massive amount of data.

> **Benefit:** This choice helped to lessen the impact of the small dataset size by using a model that is efficient and effective for the given data

---

3. Observations and Insights from Embeddings

Quality and Relevance of Semantic Relationships:

1.High Similarity Scores:

The cosine similarity matrix for random 10 words sample (provided in colab notebook) showed strong similarities between word pairs, indicating effective learning of word relationships.

> Example: High similarity between "implement" and "objective" (0.991186), "currency" and "sector" (0.993612).

2.Contextual Relationships:

> **Action-related Terms:** Words like "implement," "execution," and "objective" showed high similarity, reflecting their use in planning and strategy contexts.

> **Business and Finance Terms:** Words like "currency" and "sector" were closely related, indicating their frequent co-occurrence in business contexts.

> **Abstract Concepts:** Words such as "journey" and "established" indicated recognition of abstract relationships like progress and stability.

Embedding Structure:

> **Contextual Representation:** Words used in similar contexts had similar vector representations.

> **Vector Storage:** Embeddings were stored in a matrix, with each row representing a word and each column a dimension of the embedding vector.

Sample Embeddings:

- **Example Words:** "fold," "undertake," "will," "independent," "operational," "to," "warranted," "incidental."
- **Dimensionality:** Each word was represented by a 30-dimensional vector capturing its semantic properties.

Insights:

- **Semantic Similarity:** Words with related meanings had similar embeddings, reflecting their contextual usage.
- **Effective Learning:** The model effectively captured the semantic relationships within the dataset, as evidenced by the high similarity scores and meaningful clusters.