# Assignment 3: Word Embedding

Equal contribution.
Santiago Chevez Trejo: 848014547
Edmarck Sosa: 888977063

CPSC 488: Natural Language Processing
Dr. Christopher Ryu
23 November 2025

**Summary**

This project extends from the sentiment analysis from Assignment 2 by replacing traditional text vectorization (DTM and TF-IDF) with Word2Vec embeddings. Using the aggregated news dataset containing daily concatenated financial news per stock ticker, the goal was to generate Skip-gram and CBOW embeddings, convert each news document into a fixed length vector, and train a classifier to predict the corresponding impact score. This project evaluates whether distributed word representations provide stronger predictive performance than the earlier feature engineered approach.

**Introduction**

The dataset used for this work is the same as the sentiment analysis project, aggregated_news.csv, which involves collecting the historical prices alongside the S&P 500 index as well as the news data for sentiment analysis, which was created previously, containing three day windows of stock associated news paired with impact scores derived from market adjusted volatility. Unlike the earlier project, which used bag of words and TF IDF representations, this assignment focuses on Word2Vec embedding for the document representations that encodes the semantic meaning. Using the preprocessing functions defined in the project files of; tokenization, stopword removal, and date alignment, the news corpus was prepared for Word2Vec training and merged with the historical impact file to form labeled datasets suitable for classification.

**Part 1: Skip-gram and CBOW Embeddings**

The first portion of the project consisted of building word embeddings using “gensim” Word2Vec. The Skip gram model, using sg=1, was trained with a vector size of 100, window size 5, minimum word count 2, and 5 epochs of training. Each news article was converted into a document vector using both simple averaging and TF-IDF weighted averaging of the word embeddings. These parameters were selected to maximize the ability to capture all the semantic relationships. After training, all articles were transformed into a vector representation.

After Skipgram, a model was implemented for Continuous Bag-of-Words (CBOW). Its architecture estimates the vector representation of words by averaging the embeddings of its context words, followed by a softmax output layer of the vocabulary. Gradients were computed for the training in both input and output embedding, using the cross entropy gradient. The

resulting CBOW vectors were stored in an analogous dataset. Both embedding pipelines strictly followed the schemas enforced in “dataset_schema.py” to maintain requested CSV structure.

**Part 2: MLP Classifier**

The second part of the assignment trains a MLP classifier with PyTorch using the Skip gram and CBOW document vectors. The architecture is defined in model.py as a three layer network with ReLU activation, dropout, and a seven class output layer corresponding to shifted impact scores. Once training finishes, the section will describe the model configuration, hyperparameters, training behavior, and evaluation procedure using the datasets built in Part 1 (Table 1).

| MLP CBOW Classification Report | | | | |
|---|---|---|---|---|
| **Class** | **Precision** | **Recall** | **F1-score** | **Support** |
| 0 | 0.424 | 0.2614 | 0.3234 | 3,279 |
| 1 | 0.3319 | 0.0611 | 0.1032 | 5,056 |
| 2 | 0.2943 | 0.0139 | 0.0266 | 6,311 |
| 3 | 0.4815 | 0.9222 | 0.6327 | 25,025 |
| 4 | 0.3789 | 0.0516 | 0.0908 | 7,740 |
| 5 | 0.3347 | 0.0834 | 0.1335 | 6,090 |
| 6 | 0.4034 | 0.3755 | 0.389 | 3,683 |
| Accuracy | — | — | 0.4656 | 57,184 |
| Macro Avg | 0.3784 | 0.2527 | 0.2427 | 57,184 |
| Weighted Avg | 0.4098 | 0.4656 | 0.359 | 57,184 |

**Table 1**: This is the classification report for the CBOW MLP.

The MLP CBOW and Skip gram appear to have a relatively similar score when it comes to the classification performance, with the extended Skip gram model achieving a slightly higher accuracy. Despite class imbalance challenges, embedding models captured meaningful linguistic information and represented a more advanced and accurate approach to financial sentiment modeling (Table 2).

| MLP Skip-gram Classification Report | | | | |
|---|---|---|---|---|
| **Class** | **Precision** | **Recall** | **F1-score** | **Support** |
| 0 | 0.5408 | 0.3073 | 0.3919 | 3,749 |
| 1 | 0.469 | 0.0971 | 0.1608 | 5,924 |
| 2 | 0.6499 | 0.0379 | 0.0715 | 7,159 |
| 3 | 0.4797 | 0.9628 | 0.6403 | 27,559 |
| 4 | 0.4869 | 0.0574 | 0.1028 | 8,444 |
| 5 | 0.5052 | 0.0936 | 0.1579 | 6,745 |
| 6 | 0.5954 | 0.3461 | 0.4377 | 4,184 |
| Accuracy | — | — | 0.4877 | 63,764 |
| Macro Avg | 0.5324 | 0.2717 | 0.2804 | 63,764 |
| Weighted Avg | 0.5126 | 0.4877 | 0.3818 | 63,764 |

**Table 2**: This is the classification report for the CBOW MLP Skip gram.

**Results**

After the MLP training is completed, this section will compare classification performance between Skip-gram, CBOW, and the Assignment 2 baselines (DTM and TF-IDF). The key metrics include accuracy, precision, recall, and F1-score. The expectation is that Word2Vec embeddings capture semantic relationships within news text and potentially enable better generalization than the sparse DTM and TF-IDF methods, which do not model word meaning or context. Results will be summarized in a single comparison table showing each model's accuracy and any notable performance differences.

| Model | Accuracy | F1- score |
|---|---|---|
| DTM | 21.40% | 19.5% |
| TF-IDF | 17.73% | 18.55% |
| Curated Vectors | 21.94% | 15.88% |
| Skipgram | 48.77% | 38.18% |
| CBOW | 46.56% | 35.90% |

**Table 3:** Comparison between models from assignment 2 and Skipgram and CBOW

Table 3 summarizes the performance of all models across both assignments, allowing the comparison between them. As expected, the sparse methods used in Assigment 2,

underperformed in comparison with the ones on this assignment. Both Skipgram and CBOW have the highest accuracy. These results show that these methods can encode richer linguistic information, making them better for this task.

**Conclusion**

This project expands the sentiment analysis workflow by incorporating modern embedding techniques for financial news representation. The Skip gram and CBOW models provide richer semantic structure than the manually curated sentiment word or TF-IDF vectors used previously. When paired with a neural classifier, these embeddings allow the model to capture deeper linguistic patterns that relate news content to stock movements. Although final classification results depend on the MLP training, this embedded framework offers a more precise and accurate foundation for financial text modeling compared to earlier approaches. The MLP results and report demonstrate that word embeddings improved a far better predictive model for stock sentiment classification.