Analysis of CBOW Embeddings for Sentiment Classification

Mohammed Luqmaan Akhtar

February 18, 2026

Introduction

Computers cannot process texts unless we convert them into numbers. We have done this using one hot encoding or numerical encoding but these kinds of encodings don't preserve any semantic meaning or the relationships between words. This is why we need Embeddings. CBOW or Continous-Bag-of-Words is a method for us to train embeddings. The embedding matrix is treated as a parameter in a neural network which has to be optimised with respect to the loss function. In tradintional CBOW a softmax classifier is used to predict the proability of the next word but this is a very computation heavy process so we use negative sampling. Negative sampling creates wrong context target pairs which we can add to our loss in such a way that it maximises the probability of getting wrong targets.

Role of Subword Tokenization in Embedding Training

Word embedding models rely on a fixed vocabulary learned from the training corpus. In domain-specific text such as financial news, this creates a significant out-of-vocabulary (OOV) problem because many terms appear rarely, are newly coined, or occur in specialized forms (e.g., *underperformance*, *restructuring*, *downgraded*). When such words are treated as unknown tokens, the embedding model cannot learn meaningful representations, leading to information loss in downstream tasks like sentiment classification.

Financial language also exhibits substantial morphological variation, where related words share semantic roots but differ in surface form (e.g., *profit*, *profitable*, *profitability*; *invest*, *investment*, *investor*). Treating each variant as an independent token fragments the available training signal across multiple rare forms, weakening embedding quality. Subword tokenization addresses this by decomposing words into smaller, reusable units that capture shared morphemes and semantic components.

By representing words as compositions of subword units, embedding models can share statistical strength across related terms, reduce OOV occurrences, and generalize to unseen words. This is particularly important for CBOW training, where reliable context aggregation depends on consistent token representations. Therefore, subword tokenization forms a crucial preprocessing step for learning robust embeddings in specialized financial corpora.

Byte-Pair Encoding (BPE)

BPE is a frequency based merging algorithm. Initially the the text is broken into characters sentence wise. The characters are added to the vocabulary. Then the frequence of each pair of characters occuring adjacent to each other is counted. The pair with the most frequency is deemed to be the new token, is added to the vocabulary and in the text the parts containing that pair next to each other is merged as one. Then the process is again repeated with the new vocabulary and it continues until the desired vocabulary size is reached.

WordPiece

WordPiece tokenization is similar to BPE with the difference that instead of selecting merges purely based on raw frequency, WordPiece uses a statistical scoring function that measures how strongly two symbols are associated. At each step, the pair with the highest score is merged into a new token and added to the vocabulary.

The WordPiece merge score for a candidate pair (a, b) is defined as

$$\text{score}(a, b) = \frac{f(ab)}{f(a)\,f(b)}$$

where $f(ab)$ is the frequency of the pair occurring together, and $f(a)$ and $f(b)$ are the individual frequencies of the two symbols. This score favors merges where the joint occurrence of the pair is high relative to how often the symbols appear independently.

This formulation provides an advantage over purely frequency-based merging because it prioritizes statistically meaningful and cohesive subword units rather than simply frequent character combinations. As a result, WordPiece tends to produce linguistically consistent segments that better correspond to morphemes or stable word parts (e.g., *profit* + *##ability*, *re* + *##structuring*, BPE might have made profitability and restructuring a single token). This is also the reason why I chose WordPiece as the tokenizer in my code.

Sentiment Classifier Evaluation

Dataset Description

The sentiment classifier was evaluated on the Financial PhraseBank dataset, using the *Sentences_50Agree* split for training and *Sentences_AllAgree* for testing. Each sentence is labeled as negative, neutral, or positive based on annotator agreement. This setup provides a realistic evaluation scenario where training data contains moderate label noise while the test set contains high-confidence labels.

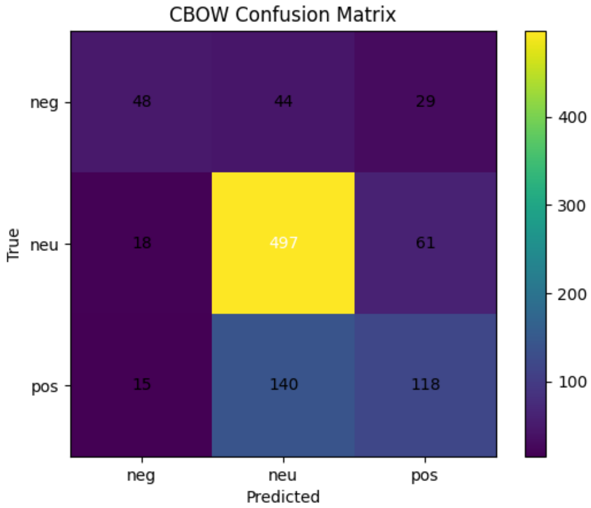
Sentence Representation Using CBOW Embeddings

Each sentence was represented by averaging the CBOW word embeddings of its constituent tokens. This produces a fixed-length dense feature vector of dimension 384 for every sentence. Averaging enables aggregation of contextual semantic information while remaining computationally efficient.

Classifier Setup

A multinomial logistic regression classifier was trained on the sentence embeddings. Performance was evaluated using macro-averaged F1 score to equally weight all sentiment classes. The CBOW-based classifier achieved a macro F1 score of 0.7723

Confusion Matrix Analysis



The confusion matrix shows that the classifier performs best on the neutral class, which dominates financial text. The most frequent errors occur between neutral and positive sentences, reflecting subtle polarity differences in financial reporting. Negative sentences are comparatively well separated due to stronger lexical cues (e.g., *decline*, *loss*, *downgrade*). Overall, the matrix indicates that CBOW embeddings capture coarse sentiment distinctions but occasionally blur mild positive and neutral statements.

Error Analysis: CBOW vs VADER

Overview of VADER

VADER is a lexicon-based sentiment analyzer. Lexicon-based sentiment analysis is a natural language processing (NLP) method that determines the emotional tone (positive, negative, or neutral) of text by matching its words against a pre-defined, labeled dictionary (lexicon) where words are assigned polarity scores.

Cases Where CBOW Outperforms VADER

CBOW embeddings outperform VADER primarily in financial-domain language where sentiment depends on context rather than isolated word polarity. For example, expressions such as *beat estimates* or *raised guidance* indicate positive sentiment despite containing neutral words. Embeddings trained on financial corpora learn these associations, whereas VADER often assigns neutral or incorrect polarity.