

# Context-Aware Synonym Replacement Using Learned Representations

Sebastian Lopez  
Texas Tech University

CS43910 – Homework 1  
Spring 2026

## 1 Introduction

The goal of this project is to design and analyze a context-aware synonym replacement system using learned representations. Rather than focusing on fluent text generation, the objective is to explore how pretrained embeddings encode meaning, how context can be approximated using similarity measures, and where such approaches fail—especially in metaphorical, poetic, or ambiguous language.

To achieve this, we implemented a multi-stage pipeline that replaces selected words in a sentence with near-synonyms while attempting to preserve sentence-level meaning. The system explicitly avoids large language models, masked language models, and rule-based synonym dictionaries, relying instead on pretrained word embeddings and sentence embeddings.

## 2 System Overview

The system follows a modular, multi-stage pipeline:

1. Sentence segmentation
2. Target word selection
3. Candidate synonym generation
4. Candidate filtering
5. Context-based scoring
6. Replacement decision

Each stage is implemented as a separate component to allow controlled experimentation and clear analysis of failure modes.

### 3 Target Word Selection

We adopt a deterministic target selection strategy: the **last content word of each sentence**. Stopwords are excluded to avoid replacing function words.

This strategy was chosen because:

- It avoids cherry-picking easy cases
- It aligns with examples provided in the assignment
- It produces consistent, repeatable behavior across test texts

## 4 Embedding Models and Rationale

### 4.1 Word-Level Embeddings

For candidate generation, we use pretrained distributional word embeddings (GloVe / Word2Vec-style vectors) loaded via `gensim`. These embeddings provide a list of semantically related words based on co-occurrence statistics.

**Rationale:**

- Efficient nearest-neighbor retrieval
- Captures semantic relatedness
- Suitable for generating candidate synonyms

### 4.2 Sentence-Level Embeddings

For context awareness, we use Sentence-BERT (SBERT) implemented in PyTorch via the `sentence-transformers` library.

**Rationale:**

- Produces sentence-level semantic representations
- Enables direct comparison between original and modified sentences
- Allows explicit reasoning about context preservation

SBERT embeddings are normalized, and cosine similarity is used as the primary scoring metric.

## 5 Candidate Generation and Filtering

Candidate words are generated as the top- $k$  nearest neighbors of the target word in the word embedding space. However, raw embedding neighbors often include unsuitable replacements. To address this, we introduce several lightweight heuristic filters (without POS taggers or external NLP libraries):

- Removal of stopwords and non-alphabetic tokens
- Plural agreement enforcement (e.g., *feathers*  $\nrightarrow$  *feather*)
- Noun-phrase context detection (e.g., “a yellow wood”)
- Blocking noun-to-adjective shifts (e.g., *wood*  $\nrightarrow$  *wooden*)
- Capitalization preservation

These heuristics do not replace embedding-based reasoning but act as grammatical guardrails that reduce trivial errors.

## 6 Context-Based Scoring

For each candidate synonym  $c$ , a modified sentence  $S_c$  is created by replacing the target word in the original sentence  $S$ . Both sentences are embedded using SBERT, and similarity is computed as:

$$\text{score}(c) = \cos(\text{SBERT}(S), \text{SBERT}(S_c))$$

The candidate with the highest similarity score is selected. Replacement occurs only if this score exceeds a predefined threshold.

## 7 Replacement Policy

The system follows a conservative replacement policy:

- Replace the target word only if the best candidate exceeds a similarity threshold
- Otherwise, retain the original word

This design prioritizes semantic preservation over aggressive substitution.

## 8 Threshold Sensitivity Study

To analyze the impact of similarity thresholds, we evaluated the system using thresholds of 0.70, 0.80, 0.83, 0.86, and 0.90 on the provided test texts (Shakespeare, Frost, Dickinson, Blake, and a haiku).

### 8.1 Observed Trends

- **Low thresholds (0.70–0.80):** Many replacements occur, but metaphorical meaning frequently collapses.
- **Mid threshold (0.83):** Metaphorical lines begin to be preserved while literal substitutions remain.
- **Conservative threshold (0.86):** Best balance between safety and usefulness.

- **High threshold (0.90):** Overly restrictive; even reasonable substitutions are blocked.

Based on these results, a threshold of **0.86** is selected as the final system configuration.

## 9 Examples

### 9.1 Successful Replacement

**Original:** “Two roads diverged in a yellow wood.”

**Modified:** “Two roads diverged in a yellow *timber*.”

This replacement preserves sentence meaning and demonstrates effective alignment between word-level similarity and sentence-level context.

### 9.2 Conservative Behavior

**Original:** “Hope is the thing with feathers.”

**Modified:** (unchanged at threshold 0.86)

The system correctly refrains from replacing the metaphorical term due to insufficient contextual similarity.

## 10 Failure Analysis

Several systematic failure modes were observed:

### 10.1 Polysemy

*players* → *teams*

In Shakespeare’s metaphor, “players” refers to theatrical roles. Word embeddings favor the sports-related sense, resulting in an incorrect substitution despite high sentence similarity.

### 10.2 Metaphor Collapse

*feathers* → *birds* (at low thresholds)

While semantically related, this replacement destroys the metaphorical meaning, illustrating that embedding similarity does not capture figurative intent.

### 10.3 Grammatical Drift

*night* → *nights*

Sentence embeddings tolerate minor inflection changes even when grammatical correctness is degraded.

## 10.4 Style and Rhythm Loss

*Splash!* → *Splashes!*

Meaning remains similar, but poetic impact and rhythm are weakened—features not captured by embeddings.

## 10.5 Minimal Context Instability

Short sentences (e.g., “Silence again.”) produce unstable substitutions due to limited contextual information.

# 11 Discussion

This project demonstrates that learned embeddings are powerful but insufficient on their own. Even with sentence-level context scoring, embedding-based similarity cannot reliably distinguish between acceptable substitutions and meaning-altering replacements in metaphorical, poetic, or ambiguous language.

These findings motivate the use of richer models and additional linguistic constraints in modern NLP systems.

# 12 Conclusion

We presented a context-aware synonym replacement system based on pretrained word and sentence embeddings. Through systematic experimentation and threshold analysis, we showed where embeddings succeed and where they fail. The results highlight both the utility and the limitations of embedding-based semantic representations and reinforce why contemporary AI systems extend beyond similarity-based approaches alone.