Information Retrieval

Phase 3: Query Expansion with Word2Vec Synonyms on the IR2025 Collection

Evaluating Word2Vec-driven Expansion on the IR2025 Collection

June 2025

Nikos Mitsakis - 3210122

Maria Schoinaki - 3210191

1. Introduction

Building on the classical **BM25** baseline ***(Phase 1)*** and the **lexical WordNet expansions** ***(Phase 2)***, this phase explores a **data-driven query expansion** strategy using **Word2Vec embeddings**. The goal is to **close vocabulary gaps** in user queries by automatically discovering **semantically similar terms** learned directly from the **IR2025** corpus itself.

In theory, it captures **distributional similarity** by learning **dense vector representations** where words sharing **contexts** are **geometrically** close in **embedding space**. **Injecting top-ranked similar terms** can thus **improve recall** for **synonymy or paraphrase cases** that **WordNet** might **not cover**.

A key aspect of this phase is our investigation into **two alternative preprocessing pipelines**:

- **Pipeline A (Preprocess → Expand):** The **standard approach** where queries are **preprocessed first** (*lowercased, tokenized, stopwords removed, stemmed*) and then **expanded** with **Word2Vec neighbors**.

- **Pipeline B (Expand → Preprocess):** An **experimental variant** where we **expand** the **raw query first** and **then apply** the **same preprocessing steps on the extended text**.

We compare these **pipelines** to assess whether **expansion** before or after **linguistic normalization** yields more **effective synonym injection**.

---

2. Data and Experimental Setup

2.1 Corpus and Index

We reuse the preprocessed **IR2025** collection and the **Elasticsearch index** configured for **BM25 similarity** and **custom English analyzers with tokenization, stopword removal, and stemming, consistent with Phases 1 and 2**.

2.2 Resources and Libraries

Key tools:

- **Gensim** (*Word2Vec implementation*)
- **scikit-learn** (*TF-IDF*)
- **NLTK** (*POS tagging and tokenization*)
- **pytrec_eval** (*official trec_eval-compatible Python metrics*)
- **ElasticSearch 8.17**

The **jsonlines library** handles input/output for corpus, queries, and results.

---

3. Word2Vec Model Training and Hyperparameter Tuning

3.1 Preprocessing

We **split** each document into **sentences, tokenize them, apply stopword removal and stemming to mirror the retrieval analyzer**. This guarantees that **semantic contexts** align with the **index representation**.

We defined a robust parameter grid:

- **vector_size: [100, 200, 300]**
- **window: [5, 8, 12]**
- **negative: [5, 10, 15]**
- **epochs: [10, 20]**
- **min_count: [2, 5]**
- **sg: [0 (CBOW), 1 (Skip-Gram)]**

We randomly sampled **30** configurations and evaluated each on:

- **MAP using a sample of 20 held-out queries.**
- **Average similarity of the top-50 high-IDF terms as a measure of vector coherence.**

The best model was selected and saved as KeyedVectors for reproducible deployment.

4. Query Expansion Method

Selection:

1. **Tokenize** and **POS-tag** the query.
2. Select **nouns** and **adjectives** only.
3. **Rank** candidates by **TF-IDF IDF score**.
4. Pick **top terms** (*n_expand=1*).

Expansion:

Retrieve **topn nearest neighbors per selected term**, **thresholded** by **similarity**, and ensure **single-word constraints**. The **final expanded query** is the **original plus the new synonyms**.

---

5. Two Preprocessing Pipelines

- **Pipeline A - Preprocess → Expand**
  - Queries undergo the **same preprocessing** as the corpus (*lowercase, stopwords, stemming*).
  - Then, we **expand** using the **trained Word2Vec model**.
  - The **expansion term** is used as is, since it **matches the preprocessed corpus style**.

- **Pipeline B - Expand → Preprocess (Experimental)**
  - **Raw query text** is **expanded first** with **semantic neighbors** in their **raw form**.
  - The **entire expanded query** is then preprocessed.

- ○ This aims to **capture more diverse neighbors before stemming** but requires **re-normalization** to align with the index.

This side-by-side test checks whether early or late expansion best preserves retrieval consistency.

---

6. Retrieval and Evaluation

For each pipeline, we:

- Saved all expanded queries to **queries_expanded_word2vec.jsonl (*Pipeline A*)** and **queries_expanded_word2vec_other.jsonl (*Pipeline B*)**.
- Issued **Elasticsearch match** searches for each expanded query at **k = 20, 30, 50**.
- Computed **MAP@k** and **avgPre@k** for **k = 5, 10, 15, 20** using **pytrec_eval**.

---

7. Results

k	Phase 1 MAP	Phase 1 avgPre@5	Phase 1 avgPre@10	Phase 1 avgPre@15	Phase 1 avgPre@20
20	0.020569	0.64	0.582	0.564	0.548
30	0.027753	0.64	0.582	0.564	0.549
50	0.039911	0.64	0.582	0.564	0.549

	Phase 2 - WordNet MAP	Phase 2 - WordNet avgPre@5	Phase 2 - WordNet avgPre@10	Phase 2 - WordNet avgPre@15	Phase 2 - WordNet avgPre@20	Phase 22 - WordNet MAP	Phase 22 - WordNet avgPre@5	Phase 22 - WordNet avgPre@10	Phase 22 - WordNet avgPre@15	Phase 22 - WordNet avgPre@20
k										
20	0.020554	0.608	0.586	0.556	0.538	0.020366	0.596	0.588	0.552	0.538
30	0.028373	0.608	0.586	0.556	0.538	0.028521	0.596	0.588	0.552	0.538
50	0.040848	0.608	0.586	0.556	0.538	0.040664	0.596	0.588	0.552	0.538

k	Phase 2 - Hypernyms MAP	Phase 2 - Hypernyms avgPre@5	Phase 2 - Hypernyms avgPre@10	Phase 2 - Hypernyms avgPre@15	Phase 2 - Hypernyms avgPre@20	Phase 22 - Hypernyms MAP	Phase 22 - Hypernyms avgPre@5	Phase 22 - Hypernyms avgPre@10	Phase 22 - Hypernyms avgPre@15	Phase 22 - Hypernyms avgPre@20
20	0.020773	0.636	0.574	0.545333	0.537	0.020761	0.64	0.576	0.546667	0.534
30	0.028601	0.636	0.574	0.545333	0.537	0.028542	0.64	0.576	0.546667	0.534
50	0.040099	0.636	0.574	0.545333	0.537	0.040006	0.64	0.576	0.546667	0.534

k	Phase 2 MAP	Phase 2 avgPre@5	Phase 2 avgPre@10	Phase 2 avgPre@15	Phase 2 avgPre@20
20	0.020552	0.608	0.586	0.556	0.538
30	0.028369	0.608	0.586	0.556	0.538
50	0.040856	0.608	0.586	0.556	0.538

k	Phase 3 MAP	Phase 3 avgPre@5	Phase 3 avgPre@10	Phase 3 avgPre@15	Phase 3 avgPre@20
20	0.021517	0.652	0.6	0.574667	0.555
30	0.028661	0.652	0.6	0.574667	0.556
50	0.040930	0.652	0.6	0.574667	0.556

k	Phase 3 MAP	Phase 3 avgPre@5	Phase 3 avgPre@10	Phase 3 avgPre@15	Phase 3 avgPre@20	Phase 33 MAP	Phase 33 avgPre@5	Phase 33 avgPre@10	Phase 33 avgPre@15	Phase 33 avgPre@20
20	0.021517	0.652	0.6	0.574667	0.555	0.021931	0.644	0.61	0.590667	0.574
30	0.028661	0.652	0.6	0.574667	0.556	0.029312	0.644	0.61	0.590667	0.574
50	0.040930	0.652	0.6	0.574667	0.556	0.042975	0.644	0.61	0.590667	0.574

---

8. Evaluation

Compare Preprocessing Pipelines within each Phase (2 & 3)

Phase 2-all Synonyms:

k	Phase 2 - WordNet MAP	Phase 2 - WordNet avgPre@5	Phase 2 - WordNet avgPre@10	Phase 2 - WordNet avgPre@15	Phase 2 - WordNet avgPre@20	Phase 22 - WordNet MAP	Phase 22 - WordNet avgPre@5	Phase 22 - WordNet avgPre@10	Phase 22 - WordNet avgPre@15	Phase 22 - WordNet avgPre@20
20	0.020554	0.608	0.586	0.556	0.538	0.020366	0.596	0.588	0.552	0.538
30	0.028373	0.608	0.586	0.556	0.538	0.028521	0.596	0.588	0.552	0.538
50	0.040848	0.608	0.586	0.556	0.538	0.040664	0.596	0.588	0.552	0.538

k	MAP (Preprocess → Expand)	MAP (Expand → Preprocess)	Δ MAP	avgPre@5	avgPre@10	avgPre@15	avgPre@20
20	0.020554	0.020366	−0.000188	0.608 → **0.596**	0.586 → **0.588**	0.556 → **0.552**	0.538 → **0.538**
30	0.028373	0.028521	+0.000148	0.608 → **0.596**	0.586 → **0.588**	0.556 → **0.552**	0.538 → **0.538**
50	0.040848	0.040664	−0.000184	0.608 → **0.596**	0.586 → **0.588**	0.556 → **0.552**	0.538 → **0.538**

- The **difference** is **marginal**: **MAP** fluctuates slightly *(~0.0001)*, with **no meaningful gain** in **early or late precision**.
- **avgPre@5** drops by **1.2 pp** across all cutoffs when using **Expand→Preprocess**, indicating that **expansion before preprocessing hurts early ranking quality with WordNet**.
- **avgPre@10-@20** remains almost **identical**, meaning the **deeper precision is unaffected.**

WordNet synonyms (est. hypernyms) tend to be more **abstract** and **less corpus-tuned**. When we **expand first** and then **preprocess** (*e.g., stemming*), we may:

- **Collapse similar forms** (*e.g., "infection" and "infect"*) into one, *or*
- **Remove stopword-like injected terms** during preprocessing.

So we have slight loss in early precision, no real gain in recall. In this case, **Preprocess→Expand remains preferable**.

Phase 2-Hypernyms:

	Phase 2 - WordNet MAP	Phase 2 - WordNet avgPre@5	Phase 2 - WordNet avgPre@10	Phase 2 - WordNet avgPre@15	Phase 2 - WordNet avgPre@20	Phase 22 - WordNet MAP	Phase 22 - WordNet avgPre@5	Phase 22 - WordNet avgPre@10	Phase 22 - WordNet avgPre@15	Phase 22 - WordNet avgPre@20
k										
20	0.020554	0.608	0.586	0.556	0.538	0.020366	0.596	0.588	0.552	0.538
30	0.028373	0.608	0.586	0.556	0.538	0.028521	0.596	0.588	0.552	0.538
50	0.040848	0.608	0.586	0.556	0.538	0.040664	0.596	0.588	0.552	0.538

Cutoff (k)	MAP (Preprocess → Expand)	MAP (Expand → Preprocess)	Δ MAP	avgPre@5	avgPre@10	avgPre@15	avgPre@20
20	0.020773	0.020761	−0.000012	0.636 → **0.640**	0.574 → **0.576**	0.545 → **0.547**	0.537 → **0.534**
30	0.028601	0.028542	−0.000059	0.636 → **0.640**	0.574 → **0.576**	0.545 → **0.547**	0.537 → **0.534**
50	0.040099	0.040006	−0.000093	0.636 → **0.640**	0.574 → **0.576**	0.545 → **0.547**	0.537 → **0.534**

- The MAP difference is **minimal** across all cutoffs (*less than ±0.0001*).
- **avgPre@5-10-15** slightly **improves** in **Expand → Preprocess**, while **avgPre@20 dips slightly**.

This suggests the preprocessing order has negligible impact for hypernym-based expansion.

Phase 3:

	Phase 3 MAP	Phase 3 avgPre@5	Phase 3 avgPre@10	Phase 3 avgPre@15	Phase 3 avgPre@20	Phase 33 MAP	Phase 33 avgPre@5	Phase 33 avgPre@10	Phase 33 avgPre@15	Phase 33 avgPre@20
k										
20	0.021517	0.652	0.6	0.574667	0.555	0.021931	0.644	0.61	0.590667	0.574
30	0.028661	0.652	0.6	0.574667	0.556	0.029312	0.644	0.61	0.590667	0.574
50	0.040930	0.652	0.6	0.574667	0.556	0.042975	0.644	0.61	0.590667	0.574

Cutoff (k)	MAP (Preprocess → Expand)	MAP (Expand → Preprocess)	Δ MAP	avgPre@5	avgPre@10	avgPre@15	avgPre@20
20	0.021517	0.021931	**+0.000414**	0.652 → 0.644	0.600 → 0.610	0.575 → 0.591	0.555 → 0.574
30	0.028661	0.029312	**+0.000651**	0.652 → 0.644	0.600 → 0.610	0.575 → 0.591	0.556 → 0.574
50	0.040930	0.042975	**+0.002045**	0.652 → 0.644	0.600 → 0.610	0.575 → 0.591	0.556 → 0.574

- **Expand → Preprocess** consistently **outperforms** in **MAP** across **all k values**.
- **Gains** are **more visible** at **higher cutoffs**, especially at **k=50** with **a +0.002 MAP increase**.
- Despite a **slight dip in avgPre@5**, **avgPre@10-20** improves consistently, **suggesting better mid-rank relevance**.
- **Word2Vec's semantic similarity** benefits from **raw**, **unprocessed input-preprocessing too early** may **remove crucial context**.

Compare all phases:

Cutoff (k)	Phase 1 (Baseline)	Phase 2A (WordNet Synonyms)	Phase 2B (WordNet Hypernyms)	Phase 3 (Word2Vec Synonyms)
MAP	0.020569 / 0.027753 / 0.039911	0.020554 / 0.028373 / 0.040848	0.020773 / 0.028601 / 0.040099	**0.021931 / 0.029312 / 0.042975**
avgPre@5	0.640	0.608	0.636	**0.644**
avgPre@10	0.582	0.586	0.574	**0.610**
avgPre@15	0.564	0.556	0.545	**0.5907**
avgPre@20	0.549	0.538	0.537	**0.574**

Phase 1 (Baseline)

- Serves as the **neutral reference**.
- **Performs decently** on **early precision** but **clearly lags in MAP at higher cutoffs**.

Phase 2A (WordNet Synonyms)

- **Slight MAP improvements** at **k=30** and **50** over **baseline**.
- **avgPre@5 drops by ~3%**, indicating **worse early precision**.
- The **inclusion of synonyms** may introduce **out-of-context** or **abstract terms**.

Phase 2B (Hypernyms)

- **Stronger MAP** gains at **k=30 (+3.06%) over baseline**.
- **Slightly better** than **2A** in **early precision** (***0.636 vs 0.608***), but still **lower than baseline**.
- **Performance plateaus or even drops slightly at higher depths**.

Phase 3 (Word2Vec Synonyms)

- **Best overall performer: wins in MAP and avgPre across nearly all cutoffs.**
- Especially **strong at k=50**, indicating **better recall and mid-rank relevance**.
- **avgPre@5 slightly lower** than **baseline** but **compensated by significant gains in avgPre@10-20**.

References

1. Elasticsearch Reference (v8.17.2) *https://www.elastic.co/guide/en/elasticsearch/reference/8.17*
2. Thakur, N., Reimers, N., Rücklé, A., Srivastava, A. and Gurevych, I., 2021. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models. arXiv preprint arXiv:2104.08663.
3. Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press.
4. Rivas, Andreia & Iglesias, Eva & Borrajo, María. (2014). Study of Query Expansion Techniques and Their Application in the Biomedical Information Retrieval. The Scientific World Journal. 2014. 1-10. 10.1155/2014/132158.
5. Fellbaum, C. ed., 1998. *WordNet: An electronic lexical database*. MIT press.
6. Bird, S., 2006, July. NLTK: the natural language toolkit. In *Proceedings of the COLING/ACL 2006 interactive presentation sessions* (pp. 69-72).
7. Rivas, A.R., Iglesias, E.L. and Borrajo, L., 2014. Study of query expansion techniques and their application in biomedical information retrieval. *The Scientific World Journal*, *2014*(1), p.132158.
8. Claudio Carpineto and Giovanni Romano. 2012. A Survey of Automatic Query Expansion in Information Retrieval. ACM Comput. Surv. 44, 1, Article 1 (January 2012), 50 pages. https://doi.org/10.1145/2071389.2071390
9. B. Xu, H. Lin and Y. Lin, "Learning to Refine Expansion Terms for Biomedical Information Retrieval Using Semantic Resources," in IEEE/ACM Transactions on Computational Biology and Bioinformatics, vol. 16, no. 3, pp. 954-966, 1 May-June 2019, doi: 10.1109/TCBB.2018.2801303.