One Word Is Not Enough: Simple Prompts Improve Word Embeddings

Rajeev Ranjan
Independent Researcher
rajeev.umd@gmail.com

Abstract

Text embedding models are designed for sentence-level applications like retrieval and semantic similarity, and are primarily evaluated on sentence-level benchmarks. Their behavior on isolated words is less understood. We show that simply prepending semantic prompts to words before embedding substantially improves word similarity correlations. Testing 7 text embedding models, including text-embedding-3-large (OpenAI), embed-english-v3.0 (Cohere), voyage-3 (Voyage AI), all-mpnet-base-v2, and Qwen3-Embedding-8B, on 3 standard benchmarks (SimLex-999, WordSim-353, MEN-3000), we find that prompts like "meaning: {word}" or "Represent the semantic concept: {word}" improve Spearman correlations by up to +0.28 on SimLex-999. Some models fail completely on bare words ($\rho \approx 0$) but recover with prompts (+0.73 improvement). Our best results achieve $\rho = 0.692$ on SimLex-999 with embed-english-v3.0 (Cohere), $\rho = 0.811$ on WordSim-353, and $\rho = 0.855$ on MEN-3000 with text-embedding-3-large (OpenAI). These results outperform classic static embeddings like Word2Vec ($\rho = 0.40$) and even the best static method LexVec ($\rho = 0.48$) on SimLex-999, establishing a new state-of-the-art for pure embedding methods. The gains extend to verb similarity (SimVerb-3500) and three downstream tasks (categorization, graded lexical entailment, synonym retrieval). This zero-shot technique requires no training and works with any text embedding model.

1 Introduction

Word embeddings revolutionized NLP when Word2Vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014) showed that words could be represented as dense vectors capturing semantic relationships. Follow-up work like fastText (Bojanowski et al., 2017) and LexVec (Salle et al., 2016) improved performance through subword information and better matrix factorization. These static embeddings were trained explicitly on word co-occurrence patterns, producing high-quality word representations.

The field has since shifted to text embedding models optimized for retrieval and RAG applications. Open-source models like Sentence-BERT (Reimers and Gurevych, 2019) use contrastive learning on datasets like SNLI (Bowman et al., 2015) and MNLI (Williams et al., 2018); commercial APIs (OpenAI, Cohere, Voyage) do not disclose their training procedures, though they are likely trained on sentence-level tasks. But what happens when we pass a single word like "dog" to these models?

This is potentially problematic for two reasons. First, isolated words may be **out-of-distribution**: if models primarily see full sentences during training, single-word inputs are unusual. Second, **tokenization may differ**: subword tokenizers often encode "dog" (sentence-initial) differently from " dog" (mid-sentence with space prefix). However, as we show, the effects of these factors are highly model-dependent.

Despite this shift, word-level embeddings remain essential for applications like word sense disambiguation, entity linking in biomedical NLP, and short-query retrieval. Yet practitioners use text embedding models for these tasks without understanding how input formatting affects quality.

We investigate a simple but effective technique: **prepending prompts to words before embedding**. For example, instead of embedding "dog" directly, we embed "meaning: dog" or "Represent the semantic concept: dog." This modification yields large improvements on word similarity benchmarks: up to +0.28 correlation improvement for well-behaved models, and +0.73 for models that fail on bare words.

Our contributions are:

*Proceedings of the 15th Joint Conference on Lexical and Computational Semantics (*SEM 2026)*, pages 464–473

1. We systematically evaluate how input formatting affects word embeddings from text embedding models, testing 8 conditions across 7 models.

2. We discover that semantic prompts significantly improve word similarity correlations, with consistent gains across all tested models.

3. We establish new state-of-the-art results for pure embedding methods on SimLex-999, WordSim-353, and MEN-3000.

2 Related Work

Contextual Word Representations. Ethayarajh (2019) showed that less than 5% of variance in BERT's contextualized representations can be explained by static embeddings, motivating research on extracting word-level representations from contextual models. Bommasani et al. (2020) proposed aggregating embeddings across multiple sentence contexts to create static word vectors. While effective, this requires a large corpus and multiple forward passes per word. Our approach instead provides a single short prompt (e.g., "meaning: dog") as a lightweight pseudo-context, achieving similar benefits with a single forward pass.

Instruction-Tuned Embeddings. INSTRUCTOR (Su et al., 2023) demonstrated that task-specific instructions improve sentence embeddings for various downstream tasks. Similarly, BGE (Xiao et al., 2023) and E5 (Wang et al., 2024) use query prefixes for retrieval tasks. However, these approaches focus on sentence-level tasks and require training with instructions. Our work shows that simple prompts work *zero-shot* for word-level similarity without any training.

Prompting Decoder LLMs for Embeddings. PromptEOL (Jiang et al., 2024) and MetaEOL (Lei et al., 2024) prompt decoder-only LLMs (e.g., LLaMA, Mistral) with templates like "This sentence: '[TEXT]' means in one word:" and extract the last-token hidden state, exploiting autoregressive next-token prediction to compress sentence meaning into a single position. Their target is sentence-level similarity (STS benchmarks). Our setup differs in three ways: (i) we use encoder-based text embedding models that produce vectors via standard pooling rather than last-token extraction, (ii) we target word-level similarity rather than sentence-level STS, and (iii) our improvement comes from shifting input closer to the training distribution of sentence-trained encoders, not from inducing summarization. The methods are not interchangeable: PromptEOL's template cannot be applied to commercial embedding APIs that do not expose hidden states.

Tokenization Effects. Garí Soler et al. (2024) examined how subword splitting affects contextualized representations, finding that average pooling works best for multi-token words. However, no prior work has systematically evaluated input *formatting* (e.g., space prefixes, semantic prompts) for word embeddings from text embedding models. We fill this gap with an evaluation across models and benchmarks.

3 Method

Given a word w, we create a prompted input $p(w)$ using a template function p. We then obtain an embedding $\mathbf{e} = f(p(w))$ where f is a text embedding model. We evaluate embeddings by computing cosine similarity between word pairs and measuring Spearman correlation (ρ) with human similarity judgments.

3.1 Prompt Conditions

We test 8 conditions organized into two categories:

Formatting conditions test tokenization effects:
- **bare**: w (baseline)
- **leading_space**: " w" (space prefix)
- **trailing_space**: "w " (space suffix)
- **both_spaces**: " w " (both)

Semantic conditions test contextual priming:
- **the_word**: "the word w"
- **word_colon**: "word: w"
- **meaning_colon**: "meaning: w"
- **instruct_semantic**: "Represent the semantic concept: w"

The formatting conditions test whether whitespace affects embedding quality. We observe that some models are sensitive to whitespace (producing different embeddings for " cat" vs "cat"), while others are insensitive (producing identical embeddings regardless of surrounding spaces). As shown in the supplementary material, all-mpnet-base-v2, BGE-large-en-v1.5, and embed-english-v3.0 show identical scores across all space conditions, indicating whitespace insensitivity. In con-

Model	Dim.	Source
Commercial APIs		
text-embedding-3-small	1536	OpenAI
text-embedding-3-large	3072	OpenAI
embed-english-v3.0	1024	Cohere
voyage-3	1024	Voyage AI
Open-Source Models		
all-mpnet-base-v2	768	Sentence-Trans.
BGE-large-en-v1.5	1024	BAAI
Qwen3-Embedding-8B	4096	Alibaba

Table 1: Text embedding models evaluated with their embedding dimensions and sources.

trast, OpenAI, Voyage, and Qwen models show varying scores, indicating whitespace sensitivity. The semantic conditions test whether providing semantic context improves the embeddings.

3.2 Models

We evaluate 7 text embedding models representing both commercial APIs and open-source options. Commercial APIs include OpenAI's text-embedding-3-small and text-embedding-3-large (OpenAI, 2024), Cohere's embed-english-v3.0 (Cohere, 2024), and Voyage AI's voyage-3 (Voyage AI, 2024). Open-source models include all-mpnet-base-v2 (Song et al., 2020; Reimers and Gurevych, 2019), BGE-large-en-v1.5 (Xiao et al., 2023), and Qwen3-Embedding-8B (Qwen Team, 2025). Table 1 summarizes the models, their embedding dimensions, and sources.

3.3 Datasets

We use four standard word similarity benchmarks:

- **SimLex-999** (Hill et al., 2015): 999 pairs rated for genuine semantic *similarity* (not relatedness). This is the most challenging benchmark as it distinguishes similar words ("car"/"automobile") from merely related ones ("car"/"road").
- **WordSim-353** (Finkelstein et al., 2002): 353 pairs measuring a mix of similarity and relatedness.
- **MEN-3000** (Bruni et al., 2012): 3000 pairs measuring semantic relatedness via crowdsourcing.
- **SimVerb-3500** (Gerz et al., 2016): 3500 verb pairs rated for verb similarity, complementing the noun-heavy benchmarks above.

Model	Bare	Best	Δ
SimLex-999			
embed-english-v3.0	0.600	**0.692**	+0.092
text-embed-3-small	0.502	0.671	+0.169
BGE-large-en-v1.5	0.568	0.650	+0.082
text-embed-3-large	0.566	0.654	+0.088
voyage-3	-0.071	0.587	+0.658
Qwen3-Embed-8B	0.286	0.567	+0.281
all-mpnet-base-v2	0.536	0.576	+0.040
WordSim-353			
text-embed-3-large	0.723	**0.811**	+0.088
BGE-large-en-v1.5	0.723	0.805	+0.082
all-mpnet-base-v2	0.744	0.761	+0.017
embed-english-v3.0	0.715	0.758	+0.043
text-embed-3-small	0.650	0.744	+0.094
voyage-3	-0.078	0.715	+0.793
Qwen3-Embed-8B	0.356	0.542	+0.186
MEN-3000			
text-embed-3-large	0.784	**0.855**	+0.071
text-embed-3-small	0.739	0.836	+0.097
voyage-3	0.096	0.826	+0.730
BGE-large-en-v1.5	0.738	0.819	+0.081
embed-english-v3.0	0.749	0.759	+0.010
all-mpnet-base-v2	0.774	0.805	+0.031
Qwen3-Embed-8B	0.439	0.708	+0.269
SimVerb-3500			
text-embed-3-small	0.460	**0.600**	+0.140
text-embed-3-large	0.510	0.590	+0.079
embed-english-v3.0	0.445	0.566	+0.121
voyage-3	-0.022	0.488	+0.510

Table 2: Word similarity results (Spearman ρ). Best prompted condition vs bare word baseline. Best overall results per dataset in bold. Model names abbreviated: text-embed = text-embedding, Embed = Embedding.

4 Results

Table 2 shows that prompts improve word similarity correlations across all models and datasets. Key findings are summarized below.

Prompts provide substantial gains. On SimLex-999, prompts improve correlations by +0.04 to +0.28 for models with reasonable baselines. Qwen3-Embedding-8B shows the largest improvement (+0.281), jumping from $\rho = 0.286$ (**bare**) to $\rho = 0.567$ with **instruct_semantic**, nearly doubling its correlation. text-embedding-3-small improves from $\rho = 0.502$ to $\rho = 0.671$ (+0.169).

Some models fail completely on bare words. voyage-3 produces near-zero or negative correlations on bare words ($\rho = -0.07$ on SimLex-999, $\rho = -0.08$ on WordSim-353, $\rho = 0.10$ on MEN) but achieves competitive results with prompts ($\rho = 0.587, 0.715, 0.826$ respectively).

This suggests voyage-3 was trained primarily on sentences and cannot meaningfully process isolated tokens without context.

Best prompts vary by model. For OpenAI's text-embedding-3 models, **instruct_semantic** and **meaning_colon** work best. For embed-english-v3.0, **instruct_semantic** is optimal on SimLex-999 while **the_word** performs best on WordSim-353. BGE-large-en-v1.5 responds best to **word_colon** across all benchmarks. Qwen3-Embedding-8B benefits most from **instruct_semantic** on SimLex-999 but from space-based formatting on WordSim-353 and MEN. This variation suggests models have different sensitivities to input formatting based on their training procedures.

Semantic prompts outperform formatting. Across all models, semantic prompts consistently outperform simple formatting changes. For text-embedding-3-small, space prefixes provide modest gains (+0.02 on SimLex), while semantic prompts like **meaning_colon** provide much larger improvements (+0.17 on SimLex). This suggests the benefit comes primarily from semantic priming rather than tokenization effects. Full results for all conditions are in the supplementary material.

Gains extend to verb similarity. On SimVerb-3500 (Gerz et al., 2016), which evaluates verb pairs rather than the noun-heavy pairs in SimLex-999/WordSim-353/MEN-3000, semantic prompts improve all four tested models. Notably, voyage-3's bare-word failure replicates ($\rho = -0.022$, essentially random), with prompts recovering correlation to $\rho = 0.488$ (+0.510). The best prompts on verbs match those for nouns (**meaning_colon**, **instruct_semantic**, **word_colon**), suggesting the effect is general rather than category-specific.

4.1 State-of-the-Art Comparison

Table 3 compares our results with prior work.[1] Note that SimLex-999 measures genuine semantic *similarity* (hard), while MEN measures *relatedness* (easier). Static embeddings learn from co-occurrence, which captures relatedness but not similarity. This explains why static methods score 0.37–0.48 on SimLex but 0.74–0.81 on MEN.

[1]Static embedding scores verified against the word-embeddings-benchmarks repository and LexVec evaluation (Salle et al., 2016).

Method	SL	WS	MEN	Type
Static Word Embeddings				
GloVe	0.37	0.52	0.74	Static
word2vec	0.44	0.70	0.74	Static
fastText	0.42	–	0.81	Static
LexVec	0.48	–	0.81	Static
Knowledge-Enhanced				
Numberbatch	0.64	0.83	0.86	+KG
Text Embed. + Prompts (Ours)				
embed-english-v3.0	**0.69**	0.76	0.80	Ours
text-embed-3-small	0.67	0.76	0.84	Ours
text-embed-3-large	0.65	**0.81**	**0.86**	Ours
BGE-large-en-v1.5	0.65	0.81	0.82	Ours

Table 3: Comparison with prior work. SL=SimLex-999, WS=WordSim-353. Static scores from Salle et al. (2016). Numberbatch uses ConceptNet (+KG). Our approach achieves new SOTA among pure embedding methods. Model names abbreviated: text-embed = text-embedding.

Among pure embedding approaches (no external knowledge), we achieve new state-of-the-art on all three benchmarks:

- **SimLex-999**: 0.69 vs 0.48 (LexVec), +44% relative improvement
- **WordSim-353**: 0.81 vs 0.70 (word2vec), +16% relative
- **MEN-3000**: 0.86 vs 0.81 (LexVec/fastText), +6% relative

ConceptNet Numberbatch (Speer et al., 2017) achieves comparable scores but requires a knowledge graph. Our approach uses only the embedding model with no additional resources. Full results for all 8 conditions are provided in the supplementary material.

4.2 Downstream Task Evaluation

To verify that improvements on intrinsic benchmarks translate to practical tasks, we evaluate on three word-level downstream tasks:

- **Word Categorization** on AP (Almuhareb and Poesio, 2004), BLESS (Baroni and Lenci, 2011), and Battig (Battig and Montague, 1969): cluster embeddings via k-means and measure purity against gold-standard categories.
- **Graded Lexical Entailment** on HyperLex (Vulić et al., 2017) (2616 pairs): Spearman correlation between cosine similarities and human hypernymy judgments.

Model	BLESS		HyperLex		Retrieval	
	Bare	Best	Bare	Best	Bare	Best
voyage-3	0.231	**0.746**	0.014	**0.325**	0.023	0.201
text-embed-3-large	0.706	0.800	0.227	0.313	0.200	**0.242**
text-embed-3-small	0.620	0.740	0.215	0.304	0.158	0.221
BGE-large-en-v1.5	0.660	0.722	0.283	0.362	0.183	0.200
embed-english-v3.0	0.699	0.699	0.229	0.329	0.176	0.176
all-mpnet-base-v2	0.737	0.746	0.271	0.297	0.158	0.179

Table 4: Downstream task results. Categorization purity on BLESS, Spearman ρ on HyperLex, MAP on WordNet synonym retrieval. Prompting improves all three tasks across nearly all models. Largest gain on voyage-3, consistent with its bare-word failure on intrinsic benchmarks.

- **Synonym Retrieval** using WordNet (Miller, 1995) (4485 queries): retrieve nearest neighbors by cosine similarity and measure mean average precision (MAP) against WordNet synonym sets.

Table 4 shows that prompting improves all three downstream tasks across nearly all models. Gains are largest for voyage-3 (e.g., +0.515 on BLESS purity, +0.311 on HyperLex), consistent with its bare-word failure on intrinsic benchmarks. For models with strong baselines like text-embedding-3-large, gains remain meaningful (+0.094 BLESS purity, +0.086 HyperLex). The best prompts for downstream tasks (**instruct_semantic**, **word_colon**, **meaning_colon**) match those for intrinsic benchmarks, confirming that our prompt recommendations transfer to practical pipelines.

5 Analysis

Why do prompts help? We identify three mechanisms. First, *training distribution mismatch*: models trained on sentences may treat isolated words as out-of-distribution. Second, *whitespace sensitivity*: BPE tokenizers (OpenAI, Voyage, Qwen) encode spaces as part of tokens, so "cat" and " cat" differ; WordPiece tokenizers (BERT-based models) strip whitespace, explaining why some models show identical scores across space conditions. Third, *semantic priming*: prompts like **instruct_semantic** signal the model to produce lexically meaningful representations, which appears most beneficial.

Model robustness varies widely. The improvement range (+0.04 to +0.73) depends on training procedures and preprocessing. Model size does not predict robustness: Qwen3-Embedding-8B (8B parameters) performs poorly on bare words, while all-mpnet-base-v2 (110M) handles them well.

Prompts induce semantic clustering. To examine the geometric effect of prompting, we measure intra-category vs. inter-category cosine similarity across six semantic categories (animals, vehicles, emotions, body parts, nature, food) using SimLex-999 words. For voyage-3 (the model with the largest gains), bare embeddings show essentially no category structure (intra=0.522, inter=0.520, separation=0.002), while **instruct_semantic** produces clear separation (intra=0.581, inter=0.470, separation=0.111). For text-embedding-3-large, intra-category similarity nearly doubles under prompting (0.388 to 0.657), indicating tighter same-category clustering. Qualitative t-SNE visualizations corroborate these metrics: bare embeddings appear scattered while prompted embeddings form visible semantic clusters.

6 Conclusion

We show that simple semantic prompts greatly improve word embeddings from text embedding models. Prepending prompts like **instruct_semantic** or **meaning_colon** yields improvements of +0.04 to +0.28 on word similarity benchmarks, with even larger gains (+0.73) for models that fail on bare words. The effect holds across noun similarity (SimLex-999, WordSim-353, MEN-3000), verb similarity (SimVerb-3500), and three downstream tasks (categorization, graded entailment, synonym retrieval). This zero-shot technique requires no training, works with any embedding model, and achieves new state-of-the-art results among pure embedding methods.

For practitioners using text embedding models for word-level semantics, adding semantic prompts to inputs is a simple way to improve results. Future work could explore prompt optimization and cross-lingual effects.

Limitations

Our study has several limitations. We evaluate only English benchmarks; cross-lingual effects remain unexplored. We test a limited set of handcrafted prompt templates; learned or optimized prompts may yield further improvements. Our SimVerb-3500 experiments cover four commercial API models; extending verb similarity eval-

uation to additional open-source models is left for future work. Commercial API models may change over time, potentially affecting reproducibility.

References

Abdulrahman Almuhareb and Massimo Poesio. 2004. Attribute-based and value-based clustering: An evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 158–165.

Marco Baroni and Alessandro Lenci. 2011. How we BLESSed distributional semantic evaluation. In *Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics*, pages 1–10.

William F. Battig and William E. Montague. 1969. Category norms of verbal items in 56 categories: A replication and extension of the Connecticut category norms. *Journal of Experimental Psychology*, 80(3p2):1.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Rishi Bommasani, Kelly Davis, and Claire Cardie. 2020. Interpreting pretrained contextualized representations via reductions to static embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4758–4781, Online. Association for Computational Linguistics.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Elia Bruni, Gemma Boleda, Marco Baroni, and Nam-Khanh Tran. 2012. Distributional semantics in technicolor. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 136–145, Jeju Island, Korea. Association for Computational Linguistics.

Cohere. 2024. Embed - Cohere documentation. https://docs.cohere.com/docs/cohere-embed.

Kawin Ethayarajh. 2019. How contextual are contextualized word representations? Comparing the geometry of BERT, ELMo, and GPT-2 embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 55–65, Hong Kong, China. Association for Computational Linguistics.

Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2002. Placing search in context: The concept revisited. *ACM Transactions on Information Systems*, 20(1):116–131.

Aina Garí Soler, Matthieu Labeau, and Chloé Clavel. 2024. The impact of word splitting on the semantic content of contextualized word representations. *Transactions of the Association for Computational Linguistics*, 12:299–320.

Daniela Gerz, Ivan Vulić, Felix Hill, Roi Reichart, and Anna Korhonen. 2016. SimVerb-3500: A large-scale evaluation set of verb similarity. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2173–2182, Austin, Texas. Association for Computational Linguistics.

Felix Hill, Roi Reichart, and Anna Korhonen. 2015. SimLex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 41(4):665–695.

Ting Jiang, Shaohan Huang, Zhongzhi Luan, Deqing Wang, and Fuzhen Zhuang. 2024. Scaling sentence embeddings with large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*.

Yibin Lei, Di Wu, Tianyi Zhou, Tao Shen, Yu Cao, Chongyang Tao, and Andrew Yates. 2024. Meta-task prompting elicits embeddings from large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

George A. Miller. 1995. WordNet: a lexical database for English. *Communications of the ACM*, 38(11):39–41.

OpenAI. 2024. Embeddings - OpenAI API. https://platform.openai.com/docs/guides/embeddings.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Qwen Team. 2025. Qwen3 embedding: Advancing text embedding and reranking through foundation models. https://arxiv.org/abs/2506.05176.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Alexandre Salle, Aline Villavicencio, and Marco Idiart. 2016. Matrix factorization using window sampling and negative sampling for improved word representations. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 419–424, Berlin, Germany. Association for Computational Linguistics.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. MPNet: Masked and permuted pre-training for language understanding. *Preprint*, arXiv:2004.09297.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. ConceptNet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pages 4444–4451.

Hongjin Su, Weijia Shi, Jungo Kasai, Yizhong Wang, Yushi Hu, Mari Ostendorf, Wen-tau Yih, Noah A. Smith, Luke Zettlemoyer, and Tao Yu. 2023. One embedder, any task: Instruction-finetuned text embeddings. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1102–1121, Toronto, Canada. Association for Computational Linguistics.

Voyage AI. 2024. Text embeddings - Voyage AI documentation. https://docs.voyageai.com/docs/embeddings.

Ivan Vulić, Daniela Gerz, Douwe Kiela, Felix Hill, and Anna Korhonen. 2017. HyperLex: A large-scale evaluation of graded lexical entailment. *Computational Linguistics*, 43(4):781–835.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024. Improving text embeddings with large language models. *arXiv preprint arXiv:2401.00368*.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. 2023. C-pack: Packed resources for general chinese embeddings. *arXiv preprint arXiv:2309.07597*.

Supplementary Material for "One Word Is Not Enough: Simple Prompts Improve Word Embeddings"

Rajeev Ranjan
Independent Researcher
rajeev.umd@gmail.com

1 Overview

This supplementary material provides complete experimental results and extended analysis for all 8 prompt conditions across 7 text embedding models and 3 word similarity benchmarks. The main paper presents aggregated findings; here we provide the full data enabling detailed model-by-model and condition-by-condition examination.

2 Complete Results

Tables 1, 2, and 3 present Spearman correlations (ρ) for each model-condition combination on SimLex-999, WordSim-353, and MEN-3000 respectively. Each table shows results for 8 prompt conditions:

- **bare**: Word only (baseline)
- **lead**: Leading space (" word")
- **trail**: Trailing space ("word ")
- **both**: Both spaces (" word ")
- **the_word**: "the word {word}"
- **word:**: "word: {word}"
- **meaning:**: "meaning: {word}"
- **instruct**: "Represent the semantic concept: {word}"

3 Detailed Analysis

3.1 Whitespace Sensitivity

A key observation from the tables is that models differ dramatically in their sensitivity to whitespace formatting. We identify two distinct behaviors:

Whitespace-Insensitive Models. Three models show *identical* scores across all whitespace conditions (bare, lead, trail, both): all-mpnet-base-v2, BGE-large-en-v1.5, and embed-english-v3.0. For example, in Table 1, all-mpnet-base-v2 scores exactly 0.536 for bare, lead, trail, and both conditions.

This insensitivity stems from tokenizer architecture. These models use **WordPiece tokenizers** that strip whitespace during preprocessing. The input "cat", " cat", "cat ", and " cat " all produce identical token sequences: `[CLS] cat [SEP]`. Thus, whitespace has zero effect on embeddings.

Whitespace-Sensitive Models. OpenAI (text-embedding-3-small/large), Voyage (voyage-3), and Qwen (Qwen3-Embedding-8B) show varying scores across whitespace conditions. These models use **BPE tokenizers** that encode leading spaces as part of the token. "cat" and " cat" produce different token IDs, affecting the resulting embeddings.

Interestingly, whitespace sensitivity can be beneficial or harmful:

- **Positive effect**: Qwen3-Embedding-8B improves from 0.286 (bare) to 0.495 (lead) on SimLex-999, a +0.21 gain from adding a space.
- **Negative effect**: text-embedding-3-large drops from 0.723 (bare) to 0.551 (lead) on WordSim-353, a -0.17 degradation.

3.2 Why Do Prompts Help?

We identify three mechanisms explaining prompt effectiveness:

(1) Training Distribution Mismatch. Text embedding models are typically trained on sentence pairs using contrastive learning. Isolated words are out-of-distribution inputs that the model has rarely or never seen during training. Prompts like "meaning: dog" transform the input into a more sentence-like format that better matches training data.

(2) Semantic Priming. Prompts explicitly signal that the model should produce a semantically meaningful representation. The **instruct** condition ("Represent the semantic concept: {word}") achieves the best results for most models, suggesting that instruction-style prompts are particularly

Model	bare	lead	trail	both	the_word	word:	meaning:	instruct
text-embed-3-small	0.502	0.523	0.532	0.550	0.594	0.597	**0.671**	0.612
text-embed-3-large	0.566	0.486	0.564	0.586	0.494	0.547	0.628	**0.654**
embed-english-v3.0	0.600	0.599	0.599	0.599	0.641	0.689	0.667	**0.692**
voyage-3	-0.071	0.086	0.143	0.391	0.289	0.426	0.457	**0.587**
all-mpnet-base-v2	0.536	0.536	0.536	0.536	0.530	**0.576**	0.569	0.510
BGE-large-en-v1.5	0.568	0.568	0.568	0.568	0.624	**0.650**	0.641	0.619
Qwen3-Embed-8B	0.286	0.495	0.419	0.474	0.450	0.432	0.530	**0.567**

Table 1: Complete SimLex-999 results (Spearman ρ) for all 8 conditions. Best per model in bold. Models showing identical scores for bare/lead/trail/both are whitespace-insensitive (WordPiece tokenizers).

Model	bare	lead	trail	both	the_word	word:	meaning:	instruct
text-embed-3-small	0.650	0.581	0.676	0.668	0.703	0.731	**0.744**	0.736
text-embed-3-large	0.723	0.551	0.739	0.724	0.757	0.752	0.774	**0.811**
embed-english-v3.0	0.715	0.715	0.715	0.715	**0.758**	0.752	0.751	0.649
voyage-3	-0.078	0.081	0.376	0.507	0.423	0.560	0.607	**0.715**
all-mpnet-base-v2	0.744	0.744	0.744	0.744	0.757	0.752	**0.761**	0.651
BGE-large-en-v1.5	0.723	0.723	0.723	0.723	0.763	**0.805**	0.772	0.711
Qwen3-Embed-8B	0.356	**0.542**	0.516	0.538	0.476	0.500	0.438	0.499

Table 2: Complete WordSim-353 results (Spearman ρ) for all 8 conditions. Best per model in bold. Note: embed-english-v3.0 performs *worse* with instruct (0.649) than bare (0.715), highlighting the need for model-specific prompt selection.

effective at activating semantic processing.

(3) Tokenization Normalization. For whitespace-sensitive models, semantic prompts may help normalize tokenization effects. When a word is embedded within a prompt, its tokenization becomes more consistent regardless of how the bare word would have been tokenized.

3.3 Model-Specific Observations

voyage-3: Extreme Sensitivity. This model shows the most dramatic behavior. Bare word correlations are near-zero or negative ($\rho = -0.071$ on SimLex, $\rho = -0.078$ on WordSim), indicating that voyage-3 cannot meaningfully process isolated words. However, with prompts, it achieves competitive results ($\rho = 0.587$ on SimLex with instruct). The improvement of +0.79 on WordSim-353 is the largest observed.

Qwen3-Embedding-8B: Best with Instruction Prompts. Despite being the largest model (8B parameters), Qwen performs poorly on bare words ($\rho = 0.286$ on SimLex). It benefits most from **instruct** on SimLex-999 but interestingly responds best to whitespace formatting on MEN-3000 (both_spaces: 0.708 vs instruct: 0.663).

all-mpnet-base-v2: Most Robust Baseline. This 110M parameter model shows the smallest improvement range (+0.02 to +0.04), indicating it already handles bare words well. It achieves its best results with **word:** on SimLex and MEN, and **meaning:** on WordSim.

BGE-large-en-v1.5: Consistent Responder. BGE shows consistent improvement with **word:** across all three benchmarks (SimLex: +0.08, WordSim: +0.08, MEN: +0.08), making it the most predictable model for prompt selection.

3.4 Dataset-Specific Patterns

SimLex-999 Shows Largest Improvements. This benchmark measures genuine semantic similarity (distinguishing "car"/"automobile" from "car"/"road"). Prompts help most here, suggesting they improve fine-grained semantic discrimination. Average improvement across models: +0.20.

MEN-3000 Shows Moderate Improvements. MEN measures semantic relatedness, which is easier than similarity. Average improvement: +0.17. Notably, embed-english-v3.0 shows minimal gain (+0.01), suggesting it already captures relatedness well.

WordSim-353 Shows Variable Results. This benchmark mixes similarity and relatedness. Some models (embed-english-v3.0) actually perform *worse* with certain prompts (**instruct**: 0.649 vs bare: 0.715), highlighting the importance of model-specific prompt selection.

Model	bare	lead	trail	both	the_word	word:	meaning:	instruct
text-embed-3-small	0.739	0.701	0.763	0.751	0.767	0.814	**0.836**	0.809
text-embed-3-large	0.784	0.665	0.795	0.797	0.768	0.767	0.794	**0.855**
embed-english-v3.0	0.749	0.749	0.749	0.749	0.756	**0.759**	0.739	0.729
voyage-3	0.096	0.116	0.399	0.520	0.528	0.672	0.672	**0.826**
all-mpnet-base-v2	0.774	0.774	0.774	0.774	0.768	**0.805**	0.765	0.737
BGE-large-en-v1.5	0.738	0.738	0.738	0.738	0.787	**0.819**	0.798	0.791
Qwen3-Embed-8B	0.439	0.634	0.633	**0.708**	0.586	0.624	0.661	0.663

Table 3: Complete MEN-3000 results (Spearman ρ) for all 8 conditions. Best per model in bold. Qwen3-Embedding-8B uniquely performs best with whitespace formatting (both: 0.708) rather than semantic prompts.

4 Practical Recommendations

Based on our analysis:

1. **Always test prompts**: Even well-performing models benefit from prompts.
2. **Start with instruct_semantic**: This works best for most models.
3. **Try word: for BERT-based models**: BGE and similar models respond well to this simpler format.
4. **Test on your target task**: Optimal prompts vary by model and benchmark.