词嵌入模型训练与评测实验报告

一、实验目标

词嵌入（Word Embeddings）作为自然语言处理中的基础技术，通过将词语映射为向量表示，能够有效捕捉词语之间的语义关系。本实验的目标是通过训练和评估英文词嵌入模型，探索不同超参数对模型性能的影响，并与预训练模型进行对比。具体目标包括：（1）训练基于 Skip-Gram 和 CBOW 的 Word2Vec 模型，评估不同参数对语义相似度和类比任务的影响，选择最优模型；（2）比较自训练模型与预训练模型（word2vec-google-news-300、glove-wiki-gigaword-300、fasttext-wiki-news-subwords-300）的性能。

二、实验思路

基于实验目标，本实验的思路分为以下几步：（1）数据预处理：首先，我们对训练集进行清洗，去除不必要的标点符号和空格，并将文本分词处理，确保数据格式规范，方便后续的模型训练；（2）模型训练：接下来，我们选用 Skip-Gram 和 CBOW 训练模型。通过调整不同的参数（包括窗口大小、词向量维度、训练周期数），我们对比了多种组合的模型表现；（3）自模型评测：模型训练完成后，我们通过语义相似度测试集（WordSim-353、SimLex-999）和类比任务（Google Analogy）对模型进行了全面评估；（4）预训练模型评测：我们采取同样的标准，对预训练模型（word2vec-google-news-300、glove-wiki-gigaword-300、fasttext-wiki-news-subwords-300）进行评估；（5）对比分析：我们通过分析不同参数对模型性能的影响，特别是窗口大小和词向量维度对相似度和类比任务的影响，进一步选择了表现最佳的自训练模型，并将最优的自训练模型与预训练模型的评测结果进行对比，总结了相关结论。（6）报告撰写：我们基于测试结果，撰写实验报告，详细展示词嵌入模型的训练过程以及不同模型在相似度和类比任务中的表现。

三、实验过程

本次实验过程包括“数据预处理”、“模型训练”、“自模型评测”、“预训练模型评测”、“对比分析”几项步骤。

（1）数据预处理。我们基于 OpenSubtitles v2016（Lison & Tiedemann，2016）英文单语语料，使用 spaCy 3.7.5（Honnibal & Johnson，2015）进行分词清洗，得到了大小为三千万 tokens 的训练语料。首先，在 OpenSubtiltles v2016 英文单语语料中，tokens 总数为 221712167。为控制本地资源占用，我们定义了生成器按句载入并处理语料。针对载入的每句语料，我们使用 spaCy 3.7.5 基于“en_core_web_sm”进行分词，筛去分词后的空白字符、标点符号，并将有效 token 以小写形式保留。对于清洗后的有效 token，为控制训练语料大小为三千万词左右，我们在得到三千万有效 token 后便停止了预处理。预处理结果以句为单位保存在 en_processed.txt 中。

（2）模型训练。我们基于 skipgram、cbow（Mikolov et al., 2013）两种方法，结合多种参数组合，训练了 36 个 Word2Vec 模型。首先，我们用 gensim 4.3.2（Řehůřek & Sojka，2010）工具包导入 Word2Vec 用于训练。在参数设置上，我们主要定义了 sg、vector_size、window、min_count、epochs。其中 sg 控制训练方式（0 为 cbow，1 为 skip-gram），vector_size 控制词向量的维度（语义学习精度），window 控制目标词的语境窗口大小（语境信息量），min_count 控制纳入词汇表的最小词频，epochs 则控制模型的训练轮数。在这些参数中，min_count 设置为 5，与 sg=[0,1]、window=[5,7,9]、vector_size=[200,300]、epochs=[5,7,10]随机结合，便得到了 36 种参数组合。基于这些参数组合，我们传入训练语料，得到了 36 种 Word2Vec 模型与词向量。后续测试将在这些模型中选出最佳模型。

（3）自模型评测。我们基于 WordSim-353（Finkelstein et al.，2002）、SimLex-999（Hill, Reichart & Korhonen, 2015）、Google analogy test set（Wikidata contributors，20189）对 36 个模型的词向量进行了测试，为相似性能力与类比能力各选择了一个最佳模型。首先，我们根据保存至本地的模型文件名重新读取了模型的各项参数，并将这些结果保存在 dataframe 中。该 dataframe 用于保存后续全部测试结果。接下来，我们分别载入 wordsim353、simlex999、google analogy 对全部模型进行测试。需要注意的是，因为自有模型均由小写语料训练，为更好地观察自有模型的学习效果，测试数据对象均转为小写。测试数据分为两类：wordsim353 和 simlex999 用于测试模型词向量的语义相似性判断能力，由词对和人工标注的相似得分构成（词 a，词 b，人工标注相似得分）。模型需要计算出词对的相似得分与人工标注得分对比；google analogy 用于测试模型词向量的类比推理能力，由两组词对构成（词 a1，词 a2，词 b1，词 b2）。模型需要推理出第二组词对的最后一个词。针对第一类相似性能力测试，我们的每个模型词向量均计算了 wordsim353 和 simlex999 中词对的相似性得分，并计算了该得分与人工标注得分的皮尔逊（r）、斯皮尔曼相关系数（rho）与两者 p 值。所得相关系数和 p 值均保留至 dataframe 中；第二类类比推理测试涉及“语义”推理和“形态”推理。因此，我们的每个模型词向量都在“语义”和“形态”的角度预测了目标词，与测试数据的答案对比，得到了语义准确率、形态准确率和总体准确率。三项准确率均保留至 dataframe 中。需要注意，在相似性能力和类比推理能力测试中，由于模型训练语料有限，测试词汇可能不在模型词表中（OOV）。针对 OOV 现象，Mikolov 等人在 Word2Vec 初始论文中测试中仅对模型词表中的测试词汇进行了测试（2013，pp. 6-7）。因此，本实验为准确反映模型所学能力，对 OOV 进行了相同处理。与此同时，为体现测试严谨性，实验还统计、报告了每种模型在每项测试中有效词汇的测试覆盖率。

得到 36 种模型的测试表现后，实验需要选出最佳模型。而在相似性判断能力和类比推理能力两种指标上，skip-gram 模型与 cbow 模型表现差异较大：skip-gram 模型总体相似性判断能力较强，类比推理能力较弱；cbow 模型反之。因此，实验决定对相似性判断和类比推理能力各选出一个最佳模型。首先，相似性能力

涉及 wordsim353 与 simlex999 两个数据集。我们分别对两个数据集的测试结果进行了排序。测试结果包括皮尔逊相关系数 r 值（与人工打分分数的相似程度）、皮尔逊 p 值（与人工打分分数相关是否显著）、斯皮尔曼相关系数 rho 值（与人工打分排名的相似程度）、斯皮尔曼 p 值（与人工打分排名相关是否显著）。排序标准如下：在皮尔逊、斯皮尔曼 p 值均小于 0.05 的情况下，优先考虑 rho 值，再考虑 r 值，从高到低排序。这是由于斯皮尔曼相关系数相较于皮尔逊相关系数对数据分布要求更低，且更能反映模型是否能捕捉到与人类感知一致的相似排序，而非数值的精确映射。在给 wordsim353 和 simlex999 结果排序后，我们取了两者前 15 名模型的交集，确保最佳模型应在两项测试上均有较好表现。在所得交集中，我们又以 simlex_rho、simlex_r、wordsim_rho、wordsim_r 值从高到低排序，选出了相似度能力最佳模型。这是由于 simlex999 比 wordsim353 更强调语义相似性（similarity）本身，而非关联性（relatedness），能更严格地反映相似度能力（Hill, Reichart & Korhonen, 2015）。而对于类比推理能力，我们对 google analogy 测试结果中的 all_accuracy 从高到低排序，得到了类比能力最佳模型。

（4）预训练模型评测。我们用自模型评测相同的方式测试了 word2vec-google-news-300、glove-wiki-gigaword-300、fasttext-wiki-news-subwords-300 模型，并保存结果。我们通过 gensim 下载了三个模型的训练词向量，并保存到本地。之后，我们用自模型测评相同的方法测试了这三个模型，将结果保存至本地。值得注意的是，在上述三模型测试中，测试数据仍然为小写。对于大小写敏感的模型，这可能造成 OOV 现象。为确保测试标准相同，OOV 处理机制和模型有效词汇覆盖率报告，可以补偿这一点。

（5）对比分析。我们基于所得最佳模型查看了窗口大小对模型性能的影响，并对比了自训练模型与预训练模型的测试结果。首先，我们分别提取了相似度任务和类比任务最佳模型的信息。针对相似度任务，我们获得了与最佳模型除 window 参数外其它参数全部相同的模型的测试结果，观察了 window 在 5、7、9 设置下，simlex_rho、simlex_r、wordsim_rho、wordsim_r 值的变化；针对类比任务，我们则以同样的方式，观察了 window 变化下语义任务准确率、词形任务准确率、整体准确率的变化。此外，我们还整合了自训练与预训练模型的测试结果，生成最终报告。

四、结果与讨论

本节将围绕我们在词向量训练与评估中的核心实验结果展开讨论，涵盖语义相似度（similarity）与类比推理（analogy）两项任务，并分析模型性能差异及参数对结果的影响。

4.1. 语义相似度任务结果分析

我们使用 WordSim-353 和 SimLex-999 两个标准数据集评估词向量模型在语义相似度任务中的建模能力，分别考察 Pearson 相关系数（r）、Spearman 等级相关系数（ρ）及词汇覆盖率（Coverage），以增强对结果的解释力和可信度。根据本文实验过程三的流程，我们得出了 Skip-gram 和 CBOW 两组模型的最优

参数，并将对应模型与三组预训练模型参数进行比较，其中 Skip-gram 模型的最优参数组合为：窗口大小 5、向量维度 300、训练轮数 5；CBOW 模型的最优参数组合为：窗口大小 5、向量维度 300、训练轮数 10（比较用数据略，参看 allmodel_info.csv）。

在语义相似度任务中，对比结果如下表所示：

model_name	wordsim-353			simlex-999		
	pearson-r	spearman-rho	coverage	pearson-r	spearman-rho	coverage
word2vec-google-news-300	0.65	0.69	0.99	0.45	0.44	1
glove-wiki-gigaword-300	0.6	0.61	1	0.39	0.37	1
fasttext-wiki-news-subwords-300	0.7	0.7	1	0.47	0.44	1
skipgram_win-5_vsize-300_ep-5	0.44	0.47	0.97	0.26	0.26	0.99
cbow_win-5_vsize-300_ep-10	0.37	0.37	0.97	0.16	0.16	0.99

根据 WordSim-353 与 SimLex-999 两个语义相似度数据集的实验结果，我们对多个预训练与自训练词向量模型进行了性能对比。分析围绕三个主要结论展开，分别从实验数据表现与结构性原因两个维度进行解释。

首先，在三个预训练模型中，fastText 与 word2vec 表现相近且最佳，GloVe 略逊一筹。如表 1 所示，fasttext-wiki-news-subwords-300 和 word2vec-google-news-300 在两个数据集上均表现出色，Spearman ρ 均达到 0.69–0.70（WordSim）和 0.44（SimLex），反映了其在语义排序任务中的强相关性。而相比之下，glove-wiki-gigaword-300 在 WordSim 上仅为 0.61，在 SimLex 上则进一步下降至 0.37，显著低于前两者。这种差异可能从模型构建方式解释：

- fastText 引入了子词建模机制（subword-level embedding）（Joulin, Grave, Bojanowski, & Mikolov, 2016），在处理形态变化或低频词时仍可生成可靠词向量，因此能比较有效地处理 OOV 词语，显著增强了语义泛化能力。
- word2vec 使用大规模 Google News 语料训练（Mikolov, Chen, Corrado, & Dean, 2013），词汇分布覆盖广泛，捕捉了丰富的上下文共现模式。
- 相比之下，GloVe（Pennington, Socher, & Manning, 2014）基于全局共现矩阵进行建模，对词序敏感度较低，在需要捕捉细粒度语义差异的任务中相对不足，这可能导致 SimLex-999 这类偏向“真正语义相似度”而非关联性的评估中表现受限。

其次，预训练模型整体优于自训练模型。从表中结果来看，无论是在 WordSim 还是 SimLex 上，三个预训练模型均显著优于我们基于本地语料训练

的 Skip-gram 与 CBOW 模型。以 SimLex 数据集为例，预训练模型的 Spearman ρ 均为 0.37 及以上，而自训练的 Skip-gram 与 CBOW 分别仅为 0.26 和 0.16，存在明显差距。这些原因可能在于：

1、语料规模与多样性不足：本地语料较小（本文仅选用 30M 词语）、主题集中（集中于电影字幕语料），难以涵盖丰富的词义变化与上下文模式；
2、训练资源受限：模型参数、训练轮数、优化器等方面无法与预训练模型相提并论；
3、词表限制影响表达能力：尽管 coverage 较高（接近 0.99），但模型对稀有或抽象词的表示质量不足，易导致整体语义空间分辨率下降。

这说明在构建通用语义模型时，大规模预训练模型在泛化能力与语义表达质量上具有显著优势。

最后，在语义相似度任务中，Skip-gram 模型的表现普遍优于 CBOW 模型。尽管最终比较表中仅展示了一个参数组合下的 Skip-gram 与 CBOW 模型，但在我们进行参数调优（如窗口大小、维度、训练轮数）过程中发现：在绝大多数参数设置下，Skip-gram 均优于 CBOW，这一趋势在 WordSim 和 SimLex 上均有体现。

该现象可归因于模型训练目标的差异：

- CBOW 通过平均上下文向量预测中心词，表达能力更偏向于高频统计分布，在建模细致的词义差异时常被“模糊化”；
- Skip-gram 独立地使用每个中心词预测其上下文，保留了更细粒度的词义表示，尤其适合语义排序类任务。

这一观察也与相关研究一致，已有实证研究表明 Skip-gram 通常在语义相似度评估中具有更强性能，特别是在训练语料不够大时尤为明显（Mikolov et. al, 2013）。

4.2. 类比任务结果分析

我们采用 Google Analogy Test Set 对各词向量模型在类比推理任务上的性能进行了评估。与 4.1.节类似，我们分别对 Skip-gram 与 CBOW 两种模型结构进行了系统参数搜索，试图寻找适用于类比任务的最佳参数组合。本轮任务对 all-accuracy 排序，得到 Skip-gram 模型的最佳参数组合为：窗口大小 5、词向量维度 200、训练轮数 5，CBOW 模型的最佳参数组合为：窗口大小 5、词向量维度 300、训练轮数 7（参看 allmodel_info.csv）。

model_name	semantics		morphology		all	
	accuracy	coverage	accuracy	coverage	accuracy	coverage
word2vec-google-news-300	0.25	0.42	0.66	0.95	0.55	0.71
glove-wiki-gigaword-300	0.77	1	0.67	1	0.72	1
fasttext-wiki-news-subwords-300	0.38	0.58	0.87	1	0.71	0.81

skipgram_win-5_vsize-200_ep-5	0.06	0.37	0.06	0.92	0.06	0.67
cbow_win-5_vsize-300_ep-7	0.08	0.37	0.15	0.92	0.13	0.67

首先，从准确率来看，三个预训练模型在语义类比（semantics）与形态类比（morphology）任务中的相对表现存在显著差异，这一点不同于它们在语义相似度任务中整体表现接近的情况。GloVe 模型在语义类比任务中表现最好，准确率达 0.77，显著优于 fastText（0.38）与 word2vec（0.25）。相反，在形态类比任务上，fastText 达到 0.87 的最高准确率，超出 word2vec（0.66）与 GloVe（0.67）。综合任务表现（All Accuracy）上，GloVe 与 fastText 分别为 0.72 和 0.71，基本持平，略优于 word2vec 的 0.55。

这种分化的表现反映出三种模型的结构优势在不同类型类比任务中的适应性。GloVe 依赖于全局共现矩阵进行建模，擅长捕捉长期共现的语义模式，这有助于处理国家、首都等类型的概念性语义类比。fastText 通过子词建模增强了对词形变化和词法规则的泛化能力，因此在形态类比中表现显著更强。而 word2vec 的表现则相对均衡但整体略逊，尤其在 semantics 上准确率偏低（0.25），可能由于其较高的 OOV 比例（coverage = 0.42）导致词项缺失影响判断。

其次，无论是 semantics、morphology 还是 overall，**三个预训练模型的表现均显著优于自训练模型**。自训练模型的准确率最高仅为 0.13，远低于预训练模型，覆盖率也稍低，虽不至于失真，但反映其词汇表示能力不足。这与我们在语义相似度任务中的结论一致，进一步验证：**自训练模型由于语料规模与上下文丰富度不足，无法有效捕捉复杂词汇关系，尤其是在类比这类结构化推理任务中表现更弱**。

最后，与语义相似度任务中 Skip-gram 明显优于 CBOW 不同，在类比任务中，**CBOW 在 morphology 和 all accuracy 上均优于 Skip-gram（0.15 vs 0.06，0.13 vs 0.06）**。这可能反映出 CBOW 对短距离上下文特征（如词形变化）更敏感，尽管语义表达力较弱，但在词法转换模式的学习上具有一定优势。然而需要指出的是，两者准确率都处于极低水平，表明本地训练语料对于类比任务支持不足，未来可以在更大规模的语料库中进行模型训练和比较。

参考文献

Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., Wolfman, G., & Ruppin, E. (2002). Placing search in context: The concept revisited. *ACM Transactions on Information Systems, 20*(1), 116–131. https://doi.org/10.1145/503104.503110

Hill, F., Reichart, R., & Korhonen, A. (2015). SimLex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics, 41*(4), 665–695. https://doi.org/10.1162/COLI_a_00237

Honnibal, M., & Johnson, M. (2015). An improved non-monotonic transition system for dependency parsing. In L. Màrquez, C. Callison-Burch, & J. Su (Eds.), *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing* (pp. 1373-1378). Association for Computational Linguistics. https://doi.org/10.18653/v1/D15-1162

Joulin, A., Grave, E., Bojanowski, P., & Mikolov, T. (2016). Bag of Tricks for Efficient Text Classification. ArXiv, abs/1607.01759.

Lison, P., & Tiedemann, J. (2016). OpenSubtitles2016: Extracting large parallel corpora from movie and TV subtitles. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016)*.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv. https://arxiv.org/abs/1301.3781

Pennington, J., Socher, R., & Manning, C. D. (2014, October). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1532-1543).

Řehůřek, R., & Sojka, P. (2010). Software framework for topic modelling with large corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks* (pp. 45–50). ELRA.

Wikidata contributors. (2018). Q32127146. Wikidata. Retrieved May 18, 2025, from https://www.wikidata.org/w/index.php?title=Q32127146&oldid=709830388