**Evaluation methods for unsupervised word embeddings**

Summary

The document discusses the evaluation methods for unsupervised word embeddings. It presents different evaluation techniques, including extrinsic and intrinsic evaluation, and proposes a model- and data-driven approach to constructing query inventories. The results of various evaluations on word embeddings are presented, showing that different embeddings perform differently in tasks. The section also discusses the importance of word frequency in evaluation and mentions the need for a comprehensive evaluation framework for word embeddings. Finally, a list of references to related papers and articles is provided.

Evaluation of word embeddings

This section of the document discusses the evaluation methods for unsupervised word embeddings. The authors present a study on the different methods used to evaluate the quality of embeddings, including extrinsic evaluation and intrinsic evaluation. They also introduce new evaluation techniques that directly compare embeddings with respect to specific queries, providing greater insight and allowing for data-driven relevance judgments. The authors propose a model- and data-driven approach to constructing query inventories and observe that word embeddings encode information about word frequency. The section goes on to discuss word embeddings and the different embedding models used in the experiments. The authors then discuss the absolute intrinsic evaluation and comparative intrinsic evaluation of relatedness using pre-collected human evaluations and a novel online user study. They present the results of the evaluation on various datasets and discuss the performance of different embedding models.

Word embedding evaluations

This section of the document discusses the results of various evaluations on word embeddings. The evaluations include intrinsic tasks such as relatedness categorization and coherence, as well as extrinsic tasks. The results show that different embeddings perform differently in these tasks, indicating that there is no one best method. The authors highlight the need for more fine-grained analysis in discerning different embedding methods.

Word embedding evaluation

This section of the document discusses the evaluation of word embeddings. The section begins by describing two tasks used to evaluate the word embeddings - noun phrase chunking and sentiment classification. The results of these tasks are then presented in Tables 4 and 5, which show the F1 scores of different word embeddings. The section goes on to discuss the results and highlight the differences between the embeddings. It also mentions the importance of considering word frequency when evaluating word embeddings. Finally, the section discusses related work in the field and concludes by emphasizing the need for a more comprehensive evaluation framework for word embeddings.

**Word Embeddings- A Survey**

Summary

The document provides a comprehensive overview of word embeddings and their importance in Natural Language Processing (NLP) tasks. The first section introduces the topic and discusses the two main types of word embeddings as well as the use of neural networks. The second section focuses on strategies for building prediction-based models, highlighting key contributions in the field. The third section provides an overview of count-based models and mentions various methods for building word embeddings. Overall, the document emphasizes the usefulness of word embeddings in NLP tasks and suggests future research directions.

Word embeddings introduction

This section of the document introduces the topic of word embeddings and their importance in Natural Language Processing (NLP) tasks. The section discusses the Vector Space Model (VSM) and statistical language modeling as background information. It also explains the two main types of word embeddings: prediction-based models and count-based models. The section mentions the use of neural networks in generating word embeddings and the notion of the distributional hypothesis, which states that words with similar contexts have similar meanings. The section also highlights the lack of comprehensive surveys on word embeddings and the usefulness of word embeddings in various NLP tasks. It concludes by stating the scope of the survey and providing an outline of the remaining sections.

Word embedding strategies.

This section provides an overview of strategies for building prediction-based models for word embeddings. It mentions different articles and their contributions to the field. Some important details include:

- Bengio et al. (2003) introduced embeddings as a by-product of training a neural network language model, and later improvements were made by Bengio and Senecal (2003) and Morin and Bengio (2005).

- Mnih and Hinton (2007) introduced the log-bilinear model and Morin and Bengio's (2005) hierarchical softmax layer was used to speed up training and testing times.

- Collobert and Weston (2008) were the first to design a model with the specific intent of learning embeddings only.

- Mikolov et al. (2013b) introduced the continuous bag-of-words (CBOW) and skip-gram (SG) models, which are log-linear models, and used negative sampling instead of hierarchical softmax.

- Bojanowski et al. (2016) suggested an improvement to the skip-gram model by learning n-gram embeddings instead of word embeddings, which is useful for languages with complex morphology.

- Count-based models, such as Latent Semantic Analysis (LSA) and the Hyperspace Analogue to Language (HAL), were also mentioned as an alternative approach to building word embeddings.

Overall, this section provides an overview of the different strategies and models used to build word embeddings and highlights key contributions in the field.

Building word embeddings.

This section of the document provides an overview of various strategies for building count-based models for word embeddings. It mentions different methods such as Singular Value Decomposition (SVD), Hierarchical Association Limit (HAL), COALS, LR-MVL, and GloVe. Each method is briefly described along with the reported gains and improvements over previous models. The section also highlights the importance of word embeddings for various Natural Language Processing (NLP) tasks and suggests future research directions.

**SGPT- GPT Sentence Embeddings for Semantic Search**

Summary

The document introduces SGPT (GPT Sentence Embeddings for Semantic Search) as a method to extract meaningful embeddings from decoders for semantic search. It discusses two settings for using decoders and presents specific implementations. The section also discusses the scaling behavior and performance of SGPT-CE model compared to the state-of-the-art BERT-based Cross-Encoder model. Results and analysis of the SGPT models on various datasets are provided, indicating state-of-the-art performance for SGPT-BE on the BEIR dataset and for SGPT-CE on semantic search tasks. The section concludes with future research directions and mentions the computational cost and licensing terms for OpenAI models.

SGPT as solution. Improved embeddings. Cross-Encoder, Bi-Encoder.

This section of the document introduces SGPT (GPT Sentence Embeddings for Semantic Search), a method to use decoder-only transformers for semantic search and sentence embeddings. It discusses the limitations of current models and proposes SGPT as a solution to extract meaningful embeddings from decoders. The section outlines two settings for using decoders: Cross-Encoder and Bi-Encoder, and introduces SGPT-CE and SGPT-BE as specific implementations. The contributions of the work include improved sentence embeddings for semantic search, the use of GPT models for search without fine-tuning, and providing alternative endpoints for search and similarity embeddings. It also provides a brief overview of related work in the field, including cross-encoders vs bi-encoders and symmetric vs asymmetric search.

Scaling and performance.

This section of the document discusses the scaling behavior and performance of the SGPT-CE model on the BEIR dataset. The SGPT-CE model is compared to the current state-of-the-art BERT-based Cross-Encoder model (BM25+CE). It is mentioned that SGPT-CE-6.1B has significantly more parameters than BM25+CE, resulting in increased latency. The re-ranking performance of SGPT-CE-6.1B is evaluated in a "Re-rank Top 10" and "Re-rank Top 100" setting, with the model's performance being bounded by the documents returned by BM25. SGPT-CE-6.1B achieves around 80% of the maximum possible performance in the "Re-rank Top 10" setting. The section also discusses the symmetric search method used for Quora dataset, where different prompts are used for smaller and larger models to improve performance. The document also includes details about the method and results of the Symmetric Search and SGPT Bi-Encoder models, including the parameters trained for different models.

SGPT performance analysis

This section of the document provides results and analysis of the SGPT (Semantic GPT) models for semantic search. The section discusses the performance of SGPT in comparison to other models on several datasets, such as USEB, Quora, STS-B, BEIR, TREC-COVID, BioASQ, and others. The results indicate that SGPT-BE (SGPT Bi-Encoder) achieves state-of-the-art performance on the BEIR dataset, outperforming other models such as cpt-text and GTR-XXL. However, it requires more storage and compute resources compared to other models. SGPT-BE is recommended for tasks where maximum performance is desired and resources are available. On the other hand, SGPT-CE (SGPT Cross-Encoder) achieves state-of-the-art performance on semantic search tasks by extracting log probabilities from pre-trained GPT models. It requires less storage but requires high availability of compute resources. The section also highlights the distinction between symmetric and asymmetric search tasks and suggests using the appropriate model based on the task classification. The conclusion suggests future research directions, such as fine-tuning GPT Cross-Encoder models on MSMARCO and exploring the combination of SGPT with GPT for generative search results. The section also mentions the computational cost and mentions the licensing terms for OpenAI models.

Bibliografia:

Liu, X., Zheng, Y., Du, Z., Ding, M., Qian, Y., Yang, Z., & Tang, J. (2021). GPT understands, too. *arXiv (Cornell University)*. <http://export.arxiv.org/pdf/2103.10385>

Muennighoff, N. (2022). SGPT: GPT Sentence embeddings for Semantic Search. *arXiv (Cornell University)*. <https://doi.org/10.48550/arxiv.2202.08904>

De M Almeida, F., & Xexéo, G. (2019). Word Embeddings: a survey. *arXiv (Cornell University)*. https://arxiv.org/pdf/1901.09069.pdf

Bakarov, A. K. (2018). A survey of Word embeddings Evaluation Methods. *arXiv (Cornell University)*. https://arxiv.org/pdf/1801.09536.pdf

Schnabel, T., Labutov, I., Mimno, D., & Joachims, T. (2015). Evaluation methods for unsupervised word embeddings. *NA*. https://doi.org/10.18653/v1/d15-1036