Interview Task Report: Chunking Strategies for IR Systems

Ivan Samarskyi
samariva(at)fit.cvut.cz

April 3, 2025

Abstract

This report presents my approach to the interview task, including a review of relevant literature, implementation details of the proposed methodology, and a presentation of the experimental results with following discussion.

1 Related Work

The research presented in "Evaluating Chunking Strategies for Retrieval" [3] is the foundation for this work. This paper provides a wide analysis of various chunking strategies for retrieval tasks, along with test corpora and implementation code for performance evaluation.

2 Methodology

My evaluation suite has an Experimenter class in the center of it, which orchestrates the entire process: chunking the corpus, generating embeddings for both chunks and queries via the Sentence Transformers [2] API, and managing embeddings and metadata in ChromaDB [4]. Chunking is handled by the FixedTokenChunker class, which creates chunks of the same token length (except for the final chunk) with optional overlap. The implementation uses the cl100k_base tokenizer from the tiktoken [1] library for text tokenization and calculating chunk start and end indices.

3 Experiments

The experimental suite is configured using `config/config.yaml` file and is managed using Hydra [5], enabling user-friendly parameter adjustments without code modifications. The main experimental parameters are as follows:

- **Chunk sizes**: A range of token counts from 100 to 1000 (100, 200, 400, 600, 800, 1000) to control splitting.
- **Overlap strategy**: A binary parameter that determines whether chunks are half-overlapping (True) or are distinct (False).
- **Retrieved chunks count**: The number of chunks to return per query (1, 3, 5, 7, 9).
- **Embedding model**: The specified embedding model name, currently set to `all-MiniLM-L6-v2`, with flexibility to use any model compatible with Sentence Transformers.

The default parameter space has 60 experimental configurations (6 chunk sizes × 2 overlap options × 5 retrieval counts), which allows extensive experimentation.

The experiments use two different embedding functions: **all-MiniLM-L6-v2** and **multi-qa-mpnet-base-dot-v1**, using "wikitexts.md" as the test corpus.

4 Results and Discussion

Figures 1 and 2 illustrate the results of the experiments described earlier.

The results demonstrate similar performance between both embedding functions, with multi-qa-mpnet-base-dot-v1 showing slightly better results, likely due to its larger embedding dimensionality compared to all-MiniLM-L6-v2 (768 versus 384).

As expected, precision peaks with minimal chunk sizes and retrieval counts, caused by the fact that most of the retrieved tokens in the chunks are relevant. Conversely, recall is maximized with larger chunk sizes and retrieval counts, capturing more relevant tokens in the extracted chunks. The IoU metric (and F1 score) prefers configurations that prioritize higher precision while maintaining acceptable recall levels. The results also show that using overlapping chunks improves performance, as it keeps important context between chunks that would otherwise be lost when splitting the text.

The complete experimental results are presented in Tables 3 and 4.

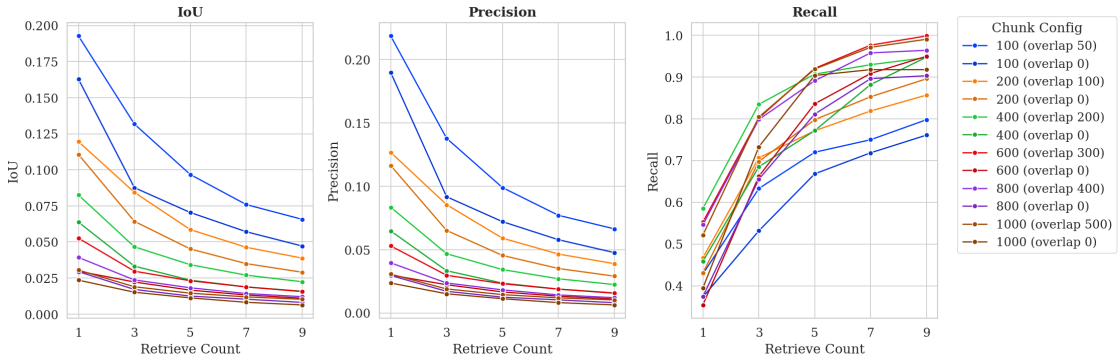

Figure 1: Comparison of evaluation metrics across different configurations using **all-MiniLM-L6-v2**

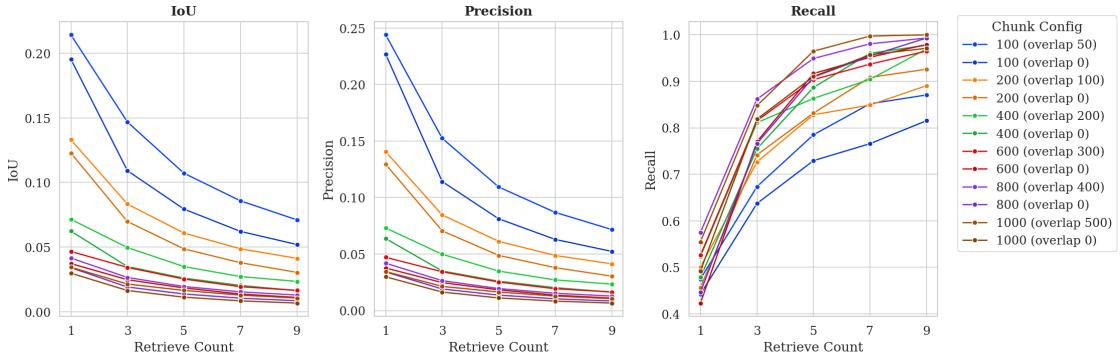

Figure 2: Comparison of evaluation metrics across different configurations using **multi-qa-mpnet-base-dot-v1**

References

[1] OpenAI. *tiktoken: OpenAI's fast BPE tokenizer*. https://github.com/openai/tiktoken. Accessed: 2025-04-03. 2022.

[2] Nils Reimers and Iryna Gurevych. "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing* (2019). URL: https://arxiv.org/abs/1908.10084.

[3] B. Smith and A. Troynikov. *Evaluating Chunking Strategies for Retrieval*. Chroma Research. 2024. URL: https://research.trychroma.com/evaluating-chunking (visited on 03/15/2024).

[4] ChromaDB Team. *ChromaDB: the AI-native open-source embedding database*. Version 0.4.22. 2024. URL: https://www.trychroma.com/.

[5] Omry Yadan. *Hydra - A framework for elegantly configuring complex applications*. Github. 2019. URL: https://github.com/facebookresearch/hydra.

Chunk Size / Overlap	IoU	Precision	Recall
100 / 0	0.085	0.092	0.61
100 / 50	**0.113**	**0.12**	0.666
200 / 0	0.057	0.058	0.735
200 / 100	0.069	0.071	0.724
400 / 0	0.031	0.031	0.749
400 / 200	0.043	0.043	0.84
600 / 0	0.019	0.019	0.742
600 / 300	0.028	0.028	**0.85**
800 / 0	0.015	0.015	0.728
800 / 400	0.021	0.022	0.831
1000 / 0	0.013	0.013	0.773
1000 / 500	0.017	0.017	0.841

Figure 3: Evaluation metrics for different configurations using **all-MiniLM-L6-v2**

Chunk Size / Overlap	IoU	Precision	Recall
100 / 0	0.1	0.108	0.678
100 / 50	**0.125**	**0.133**	0.731
200 / 0	0.062	0.063	0.773
200 / 100	0.073	0.075	0.759
400 / 0	0.032	0.032	0.812
400 / 200	0.041	0.042	0.814
600 / 0	0.021	0.021	0.808
600 / 300	0.028	0.029	0.829
800 / 0	0.017	0.017	0.814
800 / 400	0.023	0.023	0.872
1000 / 0	0.014	0.015	0.83
1000 / 500	0.019	0.019	**0.873**

Figure 4: Evaluation metrics for different configurations using **multi-qa-mpnet-base-dot-v1**