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 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.

The output of the experiments are means of the *IoU*, *precision*, and *recall* scores for each configuration.

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 for chunking with overlap only are presented in Tables 3 and 4, showing the results very similar to the ones presented in the foundational paper.

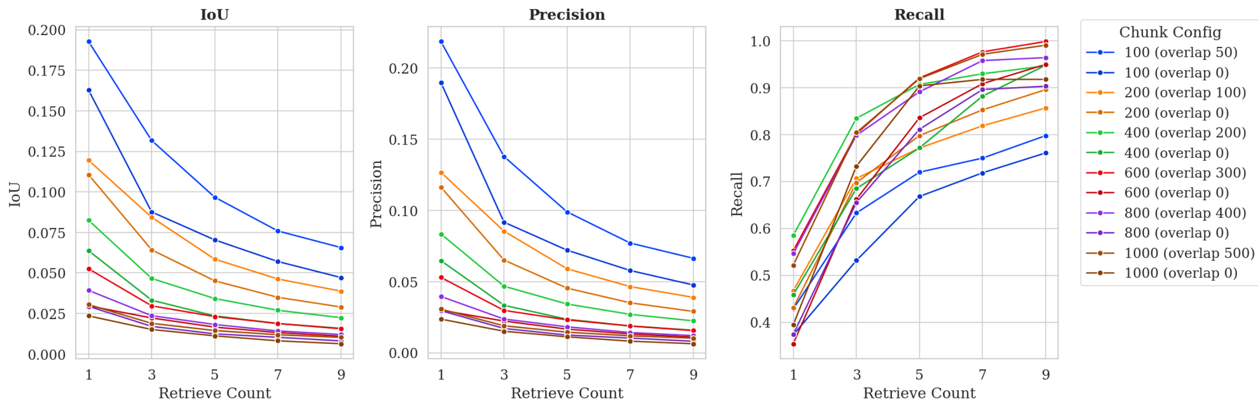

Figure 1: Comparison of evaluation metrics across different configurations using *all-MiniLM-L6-v2*

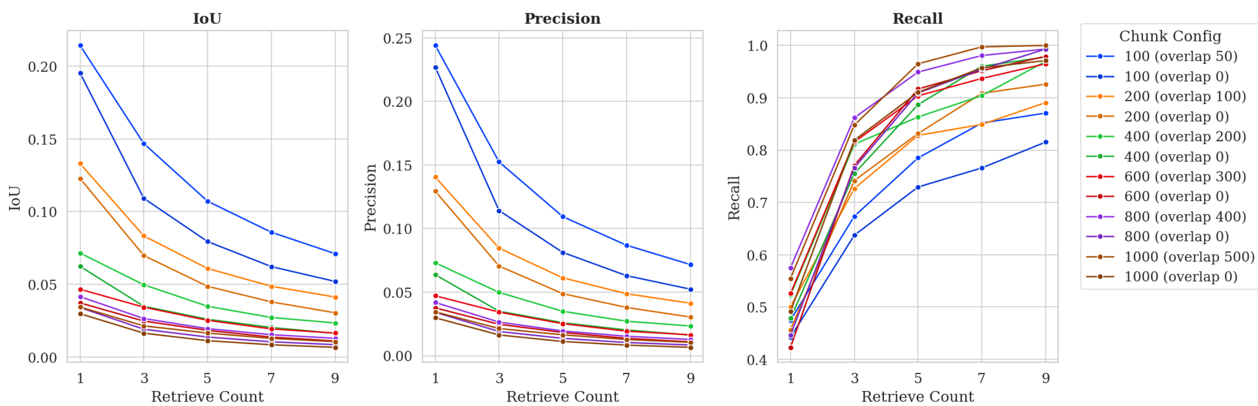

Figure 2: Comparison of evaluation metrics across different configurations using *multi-qa-mpnet-base-dot-v1*

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 / Retrieved	IoU	Precision	Recall
100 / 50 / 1	**0.214**	**0.244**	0.475
100 / 50 / 3	0.147	0.152	0.673
100 / 50 / 5	0.107	0.109	0.785
100 / 50 / 7	0.086	0.087	0.852
100 / 50 / 9	0.071	0.072	0.871
200 / 100 / 1	0.133	0.14	0.501
200 / 100 / 3	0.083	0.085	0.726
200 / 100 / 5	0.061	0.061	0.828
200 / 100 / 7	0.049	0.049	0.849
200 / 100 / 9	0.041	0.041	0.891
400 / 200 / 1	0.071	0.073	0.523
400 / 200 / 3	0.05	0.05	0.811
400 / 200 / 5	0.035	0.035	0.863
400 / 200 / 7	0.027	0.027	0.904
400 / 200 / 9	0.023	0.023	0.97
600 / 300 / 1	0.047	0.047	0.526
600 / 300 / 3	0.034	0.034	0.815
600 / 300 / 5	0.025	0.025	0.904
600 / 300 / 7	0.019	0.019	0.937
600 / 300 / 9	0.016	0.016	0.965
800 / 400 / 1	0.042	0.042	0.575
800 / 400 / 3	0.026	0.026	0.862
800 / 400 / 5	0.019	0.019	0.949
800 / 400 / 7	0.015	0.015	0.981
800 / 400 / 9	0.013	0.013	0.993
1000 / 500 / 1	0.034	0.035	0.555
1000 / 500 / 3	0.021	0.021	0.848
1000 / 500 / 5	0.016	0.016	0.965
1000 / 500 / 7	0.013	0.013	0.997
1000 / 500 / 9	0.011	0.011	**1.0**

Figure 3: Evaluation metrics for different configurations using *multi-qa-mpnet-base-dot-v1*

Chunk size / Overlap / Retrieved	IoU	Precision	Recall
100 / 50 / 1	**0.193**	**0.219**	0.428
100 / 50 / 3	0.132	0.138	0.633
100 / 50 / 5	0.097	0.099	0.72
100 / 50 / 7	0.076	0.077	0.75
100 / 50 / 9	0.066	0.066	0.798
200 / 100 / 1	0.12	0.127	0.467
200 / 100 / 3	0.084	0.085	0.706
200 / 100 / 5	0.059	0.059	0.771
200 / 100 / 7	0.046	0.046	0.818
200 / 100 / 9	0.039	0.039	0.856
400 / 200 / 1	0.082	0.083	0.585
400 / 200 / 3	0.047	0.047	0.834
400 / 200 / 5	0.034	0.034	0.907
400 / 200 / 7	0.027	0.027	0.929
400 / 200 / 9	0.022	0.022	0.946
600 / 300 / 1	0.053	0.053	0.552
600 / 300 / 3	0.03	0.03	0.801
600 / 300 / 5	0.023	0.023	0.92
600 / 300 / 7	0.019	0.019	0.976
600 / 300 / 9	0.016	0.016	**0.998**
800 / 400 / 1	0.039	0.04	0.546
800 / 400 / 3	0.024	0.024	0.798
800 / 400 / 5	0.018	0.018	0.891
800 / 400 / 7	0.014	0.014	0.957
800 / 400 / 9	0.012	0.012	0.964
1000 / 500 / 1	0.03	0.031	0.521
1000 / 500 / 3	0.019	0.019	0.804
1000 / 500 / 5	0.014	0.014	0.919
1000 / 500 / 7	0.012	0.012	0.971
1000 / 500 / 9	0.01	0.01	0.99

Figure 4: Evaluation metrics for different configurations using *all-MiniLM-L6-v2*