AnnoABSA: A Web-Based Annotation Tool for Aspect-Based Sentiment Analysis with Retrieval-Augmented Suggestions

Nils Constantin Hellwig[1], **Jakob Fehle**[1], **Udo Kruschwitz**[2], **Christian Wolff**[1]
[1]Media Informatics Group, University of Regensburg, Regensburg, Germany
[2]Information Science Group, University of Regensburg, Regensburg, Germany
nils-constantin.hellwig@ur.de, jakob.fehle@ur.de, udo.kruschwitz@ur.de, christian.wolff@ur.de

Abstract

We introduce AnnoABSA, the first web-based annotation tool to support the full spectrum of Aspect-Based Sentiment Analysis (ABSA) tasks. The tool is highly customizable, enabling flexible configuration of sentiment elements and task-specific requirements. Alongside manual annotation, AnnoABSA provides optional Large Language Model (LLM)-based retrieval-augmented generation (RAG) suggestions that offer context-aware assistance in a human-in-the-loop approach, keeping the human annotator in control. To improve prediction quality over time, the system retrieves the ten most similar examples that are already annotated and adds them as few-shot examples in the prompt, ensuring that suggestions become increasingly accurate as the annotation process progresses. Released as open-source software under the MIT License, AnnoABSA is freely accessible and easily extendable for research and practical applications.

Keywords: Annotation Tool, Aspect-Based Sentiment Analysis, Retrieval-Augmented Generation, Large Language Models, AI-Assistance

1. Introduction

Aspect-Based Sentiment Analysis (ABSA) constitutes a fine-grained approach to sentiment analysis that goes beyond document-level polarity classification by identifying specific aspects within a text and determining the sentiment orientation associated with each aspect (Pontiki et al., 2016). The field encompasses various ABSA subtasks that differ in their granularity of aspect identification. These tasks involve combinations of the following sentiment elements: aspect term *a*, aspect category *c*, opinion term *o*, and sentiment polarity *p*. For instance, in the sentence *"The pizza was delicious."*, "pizza" represents the aspect term, "food general" could constitute the associated aspect category, "delicious" serves as the opinion term, and the sentiment polarity is positive. In cases where no aspect term is given for an aspect (=implicit aspect), the aspect term is set to "NULL", e.g. *"It was delicious."*. A sentence may contain several aspects, resulting in several aspects that need to be annotated.

Due to the granularity of ABSA, the creation of annotated resources for training and evaluating ABSA-specific models remains highly time-consuming and labour-intensive (Nasution and Onan, 2024; Negi et al., 2024; Wang et al., 2023). Resources are particularly scarce for low-resource languages and domain-specific contexts (Xu et al., 2025; Ludec et al., 2023; Fehle et al., 2025).

The scarcity of annotated datasets is also reflected in the limited availability of specialized annotation tools, with only a few dedicated solutions currently existing. To date, only general-purpose annotation frameworks such as *INCEpTION*[1] (Wojatzki et al., 2017), *BRAT*[2] (Pontiki et al., 2016, 2015, 2014) and *Label Studio*[3] (Hellwig et al., 2024; Fehle et al., 2025) have been reported to be used for ABSA annotation tasks. However, these lack essential functionalities required for certain subtasks. For example, in Target Aspect Sentiment Detection (TASD), a text may contain numerous implicit aspects that do not correspond to specific textual spans but can be assigned to an aspect category and sentiment polarity. The aforementioned tools cannot handle such dynamic lists of entries, in this case aspect annotations of implicit opinions.

Given the challenges associated with manual data annotation, recent research has increasingly explored Large Language Models (LLMs) to minimize annotation effort across various NLP domains, including social media analysis (Hasan et al., 2024; Mu et al., 2024; Zhang et al., 2024), (bio-)medicine (Labrak et al., 2024; Ateia and Kruschwitz, 2023), and finance (Deußer et al., 2024; Deng et al., 2025). For ABSA, few-shot learning has demonstrated competitive performance in ABSA tasks, achieving micro-averaged F1 scores approaching those of fine-tuned models while requiring only a few examples (Hellwig et al., 2025; Zhou et al., 2024). However, these approaches still fall short of the performance levels reported for models specifically fine-tuned for ABSA tasks (Hellwig et al., 2025; Zhang et al., 2024; Zhou et al., 2024).

[1]INCEpTION (Klie et al., 2018): `https://inception-project.github.io/`
[2]BRAT: `https://github.com/nlplab/brat`
[3]Label Studio: `https://labelstud.io/`

Proceedings of the Fifteenth Language Resources and Evaluation Conference (LREC 2026), pages 7985–7998
11–16 May 2026.

In this work, we introduce **AnnoABSA**, a web-based annotation tool designed for ABSA. The tool provides extensive customization capabilities and supports all major ABSA subtasks documented in the literature (Zhang et al., 2023), including aspect term extraction, aspect category classification, sentiment polarity detection, opinion term identification, and their various combinations (pairs, triplets, quadruples).

Following recent studies in other time-intensive annotation tasks (Ghazouali and Michelucci, 2025; Kim et al., 2024; Sahitaj et al., 2025; Li et al., 2025), we aimed to integrate the capabilities of foundational models to assist annotators in the annotation process. Beyond traditional manual annotation functionality, AnnoABSA optionally incorporates a Retrieval Augmented Generation (RAG)-based suggestion mechanism that combines the strengths of LLM-based predictions with human expertise. Our proposed RAG mechanism retrieves the most semantically similar examples from the pool of instances previously annotated during the annotation process to guide the LLM in providing suggestions. This hybrid approach leverages the efficiency and consistency of large language models while preserving the nuanced judgment and domain knowledge of human annotators, thereby balancing annotation speed with quality.

Our main contributions are as follows:

- We present the first open-source annotation tool for ABSA with comprehensive compatibility across all ABSA subtasks and release it under the permissive MIT licence at https://github.com/NilsHellwig/AnnoABSA.
- We introduce a retrieval-augmented LLM-based suggestion mechanism that leverages the most similar annotated examples to enhance annotation efficiency while improving suggestion quality over time.
- We demonstrate through systematic evaluation that RAG-based suggestions significantly outperform random sampling baselines in terms of prediction performance.
- We provide evidence from a controlled study showing that expert annotators achieve a statistically significant reduction in annotation time (30.51%) when assisted by RAG-based suggestions compared to unassisted manual annotation.

2. System Description

2.1. Motivation

We present AnnoABSA, a novel annotation tool designed to address the need to support all ABSA tasks (see Appendix A) while providing an accessible and intuitive user interface (UI) that facilitates efficient annotation with minimal user interaction. In this section, we detail the system architecture, UI design decisions, and AnnoABSA's data management. Additionally, Table 1 characterizes the core features of AnnoABSA and compares them with existing annotation tools previously utilized for ABSA tasks.

2.2. System Architecture

AnnoABSA is implemented using *React.js*[4], a frontend framework specifically designed for Single-Page Applications (SPAs). We utilized *TypeScript*[5] as the primary programming language for the frontend, a statically-typed superset of JavaScript that enhances code robustness and reduces error susceptibility. The backend employs *FastAPI*[6], a Python RESTful API framework that enables the integration of essential Python packages such as *Pandas*[7] and *NumPy*[8] for efficient data processing and manipulation.

AnnoABSA can be started through a command-line interface by providing the text to be annotated in either JSON or CSV format, along with a configuration file specifying the annotation parameters.

```
$ ./annoabsa reviews.json -load-config
config.json
```

The system offers extensive customization options that can be specified in a configuration file or through CLI flags. These configuration parameters include the input text file for annotation, the sentiment elements to be considered (aspect terms, aspect categories, opinion terms, sentiment polarity), a list of valid sentiment polarities and aspect categories, boolean options for specifying if implicit aspect terms and/or opinion terms are valid. Hence, AnnoABSA supports the annotation of text data in any language. A comprehensive overview of all supported flags is provided in Appendix B. After executing the CLI tool, the AnnoABSA UI is opened in the web browser.

2.3. UI Design

The UI is presented in Figure 1. The interface components were designed for comfortable use on both desktop computers and tablets. Following minimalist design principles (Sani and Shokooh,

[4]React.js: https://reactjs.org/
[5]TypeScript: https://www.typescriptlang.org/
[6]FastAPI: https://fastapi.tiangolo.com/
[7]Pandas: https://pandas.pydata.org/
[8]NumPy: https://numpy.org/

Feature	Label Studio	INCEpTION	BRAT	AnnoABSA
Technical requirements				
Open source	√ (Apache 2.0)	√ (Apache 2.0)	√ (MIT)	√ (MIT)
Installation	Docker, Python 3	Docker, Tomcat, Java	Python 2	Python 3
Regular updates	√	√	× Oct. 2021	√
Web-based	√	√	√	√
Multi-user support	√	√	×	√
User interface & usability				
Annotation setup	XML template	Project setup	Project setup	JSON config/CLI
Interface design	Highly customizable (XML template)	Technical, functional	Technical, functional	Modern, intuitive
Language translation	√	×	×	√
Annotation guidelines	√ Configurable with XML	×	×	√ Integrated in the interface as PDF
Documentation	√	√	√	√
Support	Community forum, GitHub issues	GitHub issues	GitHub issues (inactive developers)	GitHub issues, email
Data management				
Import/Export formats	JSON, TXT, XML; CSV export only	UIMA, TSV, JSON, TXT	TXT, ANN (BRAT stand-off)	JSON, CSV
General annotation functionalities				
Multi-labelling functionality	√	√	√	√
Relationship tagging	√	√	√	√
Label customization	√ XML template	√ Project configuration	√ Config files	√ JSON configuration
Team collaboration	√ Full support with role management	×	×	×
Token identification[1]	√	√	√	√
ABSA-specific features				
Validation[2]	×	×	×	√
AI-based suggestions	√ Integration for individual models/APIs	√ Integration for individual models/APIs	×	√
Dynamic lists[3]	×	×	×	√
Assigning (multiple) categories to a text span	×	×	×	√

Table 1: **Comparison of Annotation Tools for ABSA.** Detailed comparison of four annotation platforms across technical requirements, usability features, data management capabilities, and ABSA-specific functionalities. The table constitutes an extension of the comparison provided by Colucci Cante et al. (2024, p. 359), who evaluated non-semantic textual annotation tools based on the following criteria: multi-labelling functionality, annotation suggestions, relationship tagging, label customization, and team collaboration.

[1] *Token Identification*: Automatic identification of token boundaries in text to prevent annotation errors where spans begin or end within words.
[2] *Validation*: Label Studio, INCEpTION, and BRAT do not enforce mandatory linking of sentiment elements within aspect tuples. For example, in the TASD task (tuples consisting of aspect term, aspect category, and sentiment polarity), annotators may mark an aspect term but forget to set the corresponding aspect category and sentiment polarity. AnnoABSA prevents such errors through strict validation, ensuring each tuple contains exactly the required number of sentiment elements for a specific task. Individual sentiment elements are also validated (e.g., sentiment polarity must be "positive", "negative", or "neutral"; aspect categories must be from predefined lists; aspect/opinion terms must be either NULL for implicit aspects/opinions or substrings of the text that should be annotated).
[3] *Dynamic Lists*: Annotations for ABSA tasks may comprise an unlimited number of aspects. As an example, for the ACD task (which considers pairs of aspect category and sentiment polarity), a tool should offer the functionality to assign unlimited combinations of aspect categories and sentiment polarities to documents. Label Studio, INCEpTION, and BRAT cannot handle unlimited assignments of categorical variable combinations to documents.

2016), we focused on the main content, avoided redundant visual elements, and employed flat design aesthetics.

2.3.1. Topbar

The topbar provides a convenient navigation between examples, using arrow keys to move to the previous or next instance. Additionally, a double-

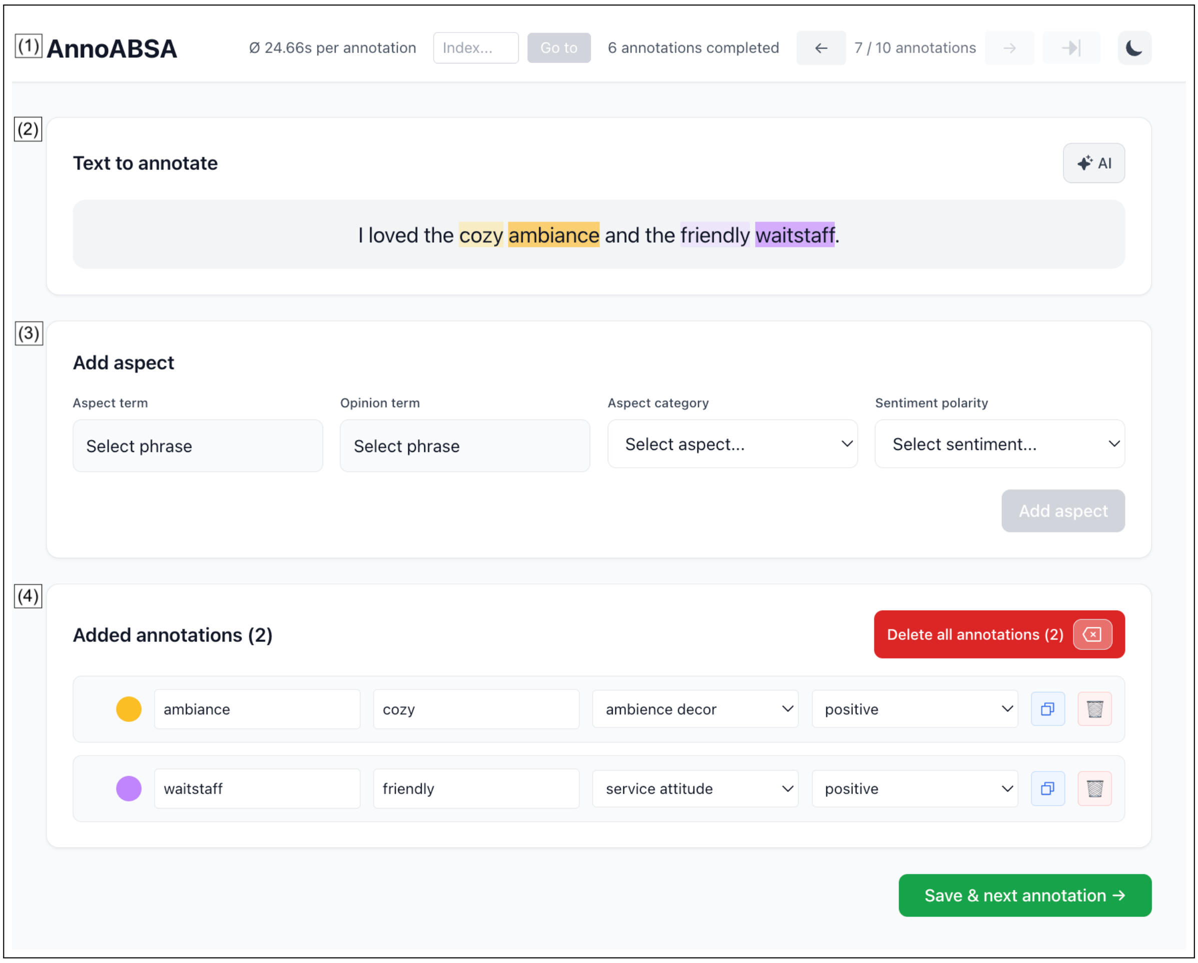

Figure 1: **UI of AnnoABSA.** The UI consists of four components: (1) top navigation bar to change the currently displayed example, (2) annotated text with highlighted sentiment annotations, (3) aspect addition panel, and (4) editable list of added annotations.

arrow button on the right allows users to jump directly to the unannotated example with the highest index, and an input field enables navigation to a specific index.

2.3.2. Form to Add New Aspect

The subsequent section in the UI enables adding new aspect annotations. Depending on the CLI specification, varying numbers of sentiment elements to be annotated are displayed. Figure 1 shows all four elements, though configurations with fewer sentiment elements can also be configured. Categorical sentiment elements, aspect category and sentiment polarity, can be selected using drop-down menus, while aspect term and opinion term each trigger a popup interface.

As shown in Figure 2, the popup displays the text to be annotated twice, once for aspect term annotation and once for opinion term annotation. When only one phrase type needs to be annotated, the text is displayed once. We considered implementing a tool-switching approach (separate selectable tools for aspect term and opinion term selection), but this would have required additional clicks. Our chosen interface requires only phrase marking and clicking "Done", thus minimizing user actions to only those absolutely necessary.

Phrase boundaries are defined by clicking on individual characters or, if configured, tokens. Once a phrase is marked, both the text and position are displayed for confirmation. Notably, this popup appears both when annotating new aspects and when editing existing ones. After all sentiment elements are specified, users click "Add aspect" to append the sentiment element to the list of applied annotations presented below the "Add aspect" section.

2.3.3. Annotation List

All annotated aspects are displayed in a list and can be modified at any time. We included a duplication button for aspects, designed to assist in cases where aspects differ in only one sentiment element. For example, in the sentence *"The pizza*

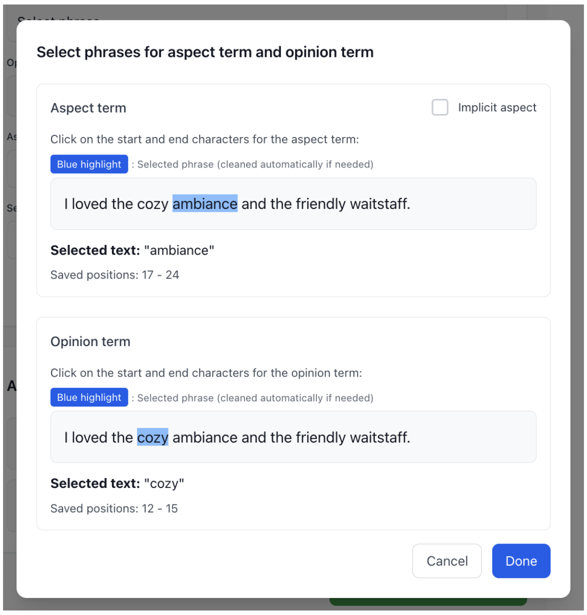

Figure 2: **Popup to add or manipulate phrase annotations.** In case both aspect term and opinion term annotations are required, the text is displayed twice, once for each phrase annotation. Implicit aspects can be marked using a checkbox.

and the burger were delicious", the Aspect Sentiment Quad Prediction (ASQP) gold label consists of two quadruples: `('pizza', 'food quality', 'delicious', 'positive')` and `('burger', 'food quality', 'delicious', 'positive')`. During annotation, one quadruple can be duplicated and the relevant sentiment element modified accordingly. Annotations can be deleted as needed.

2.4. Data Handling

The CSV or JSON file containing all examples to be annotated is directly modified during the annotation process, with annotations being appended as the annotation work progresses. In addition to the sentiment elements and phrase positions within the given text, annotation duration can optionally be stored. The use of JSON format enables straightforward integration for NoSQL databases if required, with minimal code modifications.

3. Retrieval Augmented Suggestions

Given the absence of annotated training examples for supervised models in a situation where a human annotator starts annotating text for ABSA, we adopt an LLM-based approach for generating annotation suggestions. Previous research (Hellwig et al., 2025; Zhou et al., 2024; Zhang et al., 2024) demonstrated that LLMs utilizing a small, fixed set of few-shot examples achieve performance approaching fine-tuned models, particularly in low-resource scenarios, and, that Retrieval Augmented Generation (RAG)-based few-shot learning achieved higher performance scores than random sampling across various ABSA tasks.

This section examines technical considerations including model support and prompting techniques, followed by a comparative evaluation of random and RAG-based sampling approaches for annotation suggestion generation.

3.1. Model Support

AnnoABSA provides flexible model integration through appropriate CLI parameters, supporting both commercial models via the OpenAI API[9] and open-source models through the locally hosted Ollama API[10]. These toolkits were selected based on their platform independence and native implementation of structured output capabilities. Structured outputs via guided decoding ensure that LLM suggestions are constrained to the aspect categories and sentiment polarities defined in the CLI configuration, preventing the generation of hallucinated labels. Furthermore, structured outputs guarantee that predicted phrases are present in the text that needs to be annotated.

3.2. Prompt

We adopted the prompt structure employed by Hellwig et al. (2025) and Gou et al. (2023), which comprises a task description, demonstrations, and the text to be annotated. A modification of the prompt involves formatting the demonstration's labels in JSON, similar to the aforementioned enforced structured output. An example of the prompt template is provided in Appendix C.

3.3. Demonstration Selection Strategy Evaluation

Although the RAG-based approach by Zhou et al. (2024) which utilized the k most semantically similar training examples as few-shot demonstrations achieved higher performance scores than random selection, their methodology cannot be directly transferred to the annotation process. Their approach supposes access to a fully annotated training corpus for example selection, which contrasts with annotation scenarios where the set of human-annotated instances (pool) expands during the labelling process. To address this discrepancy, we conducted a performance analysis between random sampling and RAG-based sampling strategies within an annotation framework. All LLM executions in our performance analysis were conducted

[9]OpenAI API: `https://openai.com/api/`
[10]Ollama: `https://ollama.com/`

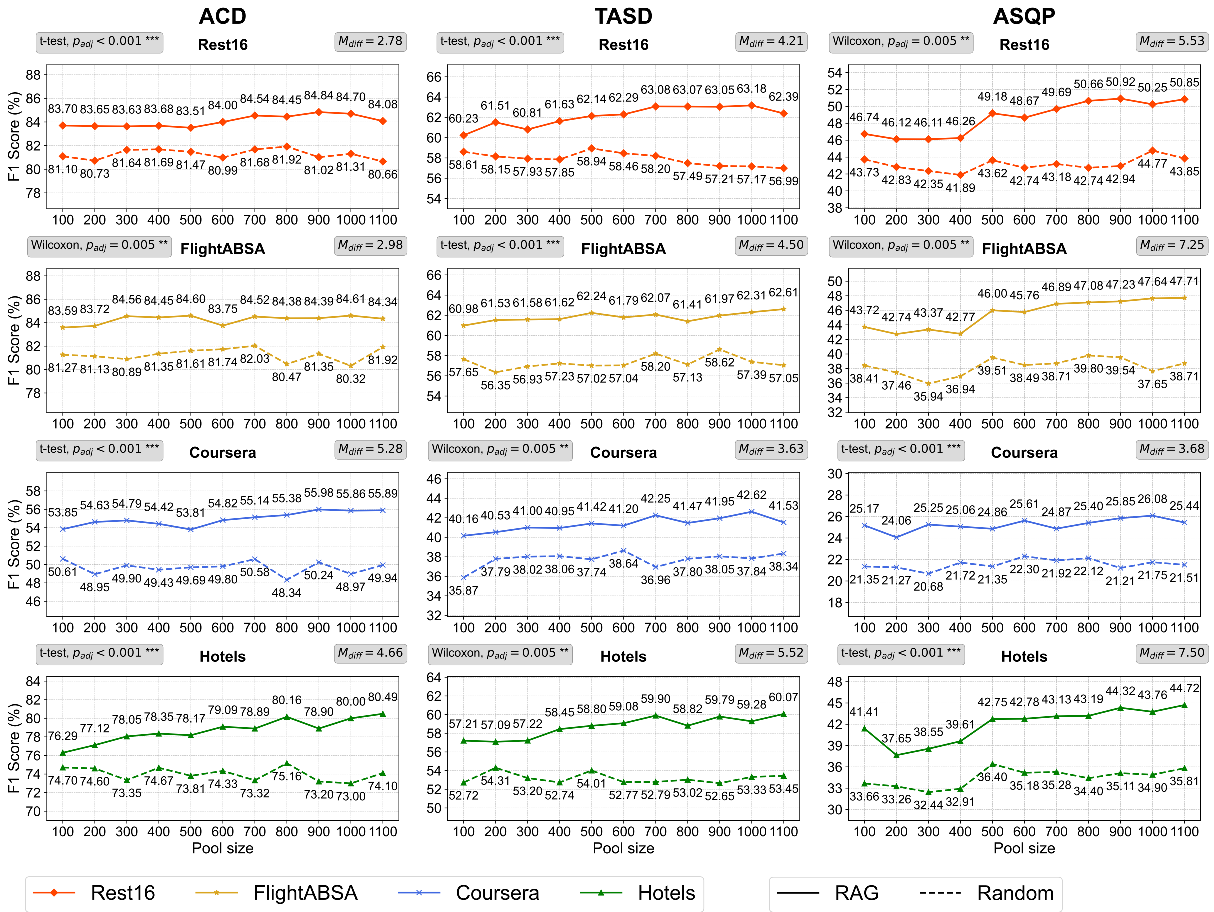

Figure 3: **F1 score comparison of RAG-based and random sampling approaches.** RAG consistently outperforms random sampling across all configurations. Statistical tests (paired t-test or Wilcoxon signed-rank test with Holm-Bonferroni correction) confirm all differences are significant. M_{diff} shows mean performance differences.

on an NVIDIA RTX PRO 6000 GPU with 96 GB VRAM.

3.3.1. Methodology

Selection Strategy. Following the methodology established by Zhou et al. (2024), we employ BM25 (Robertson and Zaragoza, 2009) as a sparse retrieval algorithm for RAG, as it enables rapid similarity comparisons, thereby facilitating fast suggestion generation. For random sampling, the examples are randomly selected from the pool.

Pool Size. To investigate performance evolution as the number of gold-labelled examples available for few-shot demonstration retrieval increases, we analysed a spectrum ranging from 0 to 1,100 available training examples in incremental steps of 100. For instance, when the pool size is 300, the few-shot examples are selected from 300 examples. We considered 1,100 examples as the maximum pool size, as this was the largest multiple of 100 available across all datasets.

Datasets. Performance was evaluated across four datasets spanning diverse review domains. These datasets encompass reviews on restaurants (SemEval 2016, Rest16) (Pontiki et al., 2016; Zhang et al., 2021), e-learning courses (Coursera) (Chebolu et al., 2024), airlines (FlightABSA) (Hellwig et al., 2025), and hotels (Chebolu et al., 2024). We randomly selected 1,100 examples from the respective training sets.

LLM Configuration. Similar to Hellwig et al. (2025), we employed *Google*'s Gemma-3-27B (Team et al., 2025) with the temperature set to 0, ensuring deterministic selection of the highest probability token during next-token prediction. The prompt context incorporated 10 few-shot examples. For each combination of pool size, task and dataset, the LLM was executed five times. Each time, a different random seed was employed to ensure varied selections of the 1,100 examples extracted from the respective training set.

Tasks. We evaluated three tasks of varying complexity: one single-aspect task (Aspect Category Detection, ACD) and two tuple prediction tasks: Target Aspect Sentiment Detection (TASD), and

Aspect Sentiment Quad Prediction (ASQP). ACD requires the identification of all aspect categories addressed within a given text. TASD extracts opinion triplets comprising aspect term, aspect category, and sentiment polarity, while ASQP additionally extracts opinion terms, representing the most fine-grained ABSA task.

Evaluation metrics. Evaluation for each step was performed on the full respective test set of each dataset. As common in the field of ABSA, the reported evaluation metric is the micro-averaged F1 score (Zhang et al., 2023). We publish all predicted labels and provide performance scores of the macro-averaged F1 score, precision, and recall in our GitHub repository.

3.3.2. Results & Discussion

The results (see Figure 3) demonstrated that the RAG approach consistently outperformed random few-shot selection across all pool sizes and tasks for all datasets. Notably, RAG achieved substantial performance gains, with differences (M_{diff}) of up to six percentage points compared to random sampling in several instances.

Pool	LLM Execution Time (Seconds)					
	ACD		TASD		ASQP	
	RAG	Random	RAG	Random	RAG	Random
100	0.816	0.851	1.292	1.240	1.708	1.622
200	0.848	0.852	1.291	1.204	1.723	1.626
300	0.846	0.871	1.291	1.211	1.736	1.642
400	0.862	0.856	1.298	1.221	1.743	1.622
500	0.858	0.867	1.314	1.220	1.743	1.647
600	0.868	0.852	1.308	1.219	1.745	1.630
700	0.864	0.884	1.311	1.210	1.739	1.640
800	0.865	0.872	1.320	1.210	1.755	1.638
900	0.868	0.864	1.322	1.204	1.759	1.635
1,000	0.872	0.856	1.310	1.202	1.745	1.636
1,100	0.866	0.859	1.296	1.199	1.756	1.617
AVG	0.858	0.862	1.305	1.213	1.741	1.632

Table 2: **LLM inference time comparison for ABSA tasks.** Average execution time per prediction (in seconds) comparing RAG-based versus random sampling strategies across varying pool sizes for ACD, TASD, and ASQP tasks.

Statistical significance was tested using paired t-tests for normally distributed differences and Wilcoxon signed-rank tests for non-normally distributed sets, followed by Holm-Bonferroni correction (Holm, 1979) for multiple testing across 12 comparisons ($\alpha = 0.05$). For each task-dataset combination, we compared the 11 performance scores (one at each pool size) obtained under RAG versus random sampling. Statistical significance was observed across all task-dataset combinations.

Overall, our findings demonstrated that a RAG-based approach achieved higher performance scores than random sampling in the context of a growing pool from which few-shot examples are drawn, with statistically significant differences observed across all tasks and datasets. Accordingly, the RAG approach was integrated into the final version of AnnoABSA.

Further performance improvements could potentially be achieved by incorporating a larger number of few-shot examples, as previously demonstrated for random sampling by Hellwig et al. (2025). However, such improvements would come at the cost of increased prediction latency, as additional tokens would need to be loaded into the model context. On our NVIDIA RTX PRO 6000 hardware, predictions for individual tasks were completed within seconds, as shown in Table 2, but increasing the number of few-shot examples would proportionally extend processing time or financial costs in the case of a commercial LLM.

3.4. User Study on Annotation Speed

As with other LLM-assisted annotation tools, the main motivation behind AnnoABSA's LLM-based suggestions is to reduce annotation time (Kim et al., 2024). We therefore conducted a user study to evaluate whether AnnoABSA enables faster annotation compared to manual annotation without AI assistance. We selected ASQP as the target task, which considers the most sentiment elements per tuple among all ABSA tasks (aspect term, aspect category, opinion term, and sentiment polarity) and is therefore the most comprehensive annotation task.

3.4.1. Methodology

We conducted a within-subjects user study with 8 expert annotators, comprising four PhD students and four master's students in computer science, all with prior experience in ABSA annotation tasks. The study was conducted in a controlled usability laboratory environment to minimize external distractions, with a research supervisor available exclusively during the initial briefing phase to answer questions before the annotation tasks commenced.

The study employed a counterbalanced design, where each participant completed two annotation sessions on separate days to eliminate fatigue effects: one session with AI suggestions enabled and one without suggestions. To control for potential ordering and learning effects, we implemented Latin square counterbalancing, systematically varying both the system order (AI-first vs. baseline-first) and dataset assignment (subset A vs. subset B) across participants.

Subsets A and B each consisted of 50 randomly sampled examples from the restaurant dataset published as part of SemEval 2016 Task 5 (Pontiki et al., 2016). Participants were provided with the corresponding annotation guidelines for reference. Prior to annotating the 50 test examples in each session, participants completed a familiarization phase with 5 demonstration examples to become acquainted with the tool interface, during which they could ask questions and receive guidance from the research supervisor.

We employed the same LLM (Gemma-3-27B) and GPU configuration as used in our selection strategy evaluation. Given that the time annotators took per example served as our primary evaluation metric, we implemented frontend-based time tracking. The timer was started upon loading each example and terminated when participants opened the subsequent example in the interface.

3.4.2. Results & Discussion

A paired t-test revealed a statistically significant difference in mean annotation time per example between the AI-assisted condition ($M = 24.19$, $SD = 3.49$) and the baseline condition ($M = 34.80$, $SD = 6.92$), $t(7) = -3.640$, $p = 0.008$. The Shapiro-Wilk test confirmed that the distribution of differences satisfied the normality assumption ($W = 0.944$, $p = 0.647$). These findings demonstrate that LLM-generated suggestions can substantially accelerate human annotation workflows, yielding a 30.51% reduction in annotation time.

Our results align with prior work demonstrating that AI-assisted annotation can reduce the temporal demands of human labelling tasks (Kim et al., 2024; Ni et al., 2024; Sahitaj et al., 2025; Li et al., 2025), though the reduction observed in our study is more modest than that reported for other tasks. Notably, Ni et al. (2024) demonstrated that nearly half of factual claim annotations could be fully automated through consistency-checked GPT-4 outputs, thereby reducing expert annotation effort by approximately 50%. Similarly, Sahitaj et al. (2025) reported a 73% reduction in annotation time for propaganda detection in tweets, while Li et al. (2025) observed a 36% reduction for AI-assisted pre-annotation of x-ray images.

4. Conclusion & Future Work

We presented AnnoABSA, the first open-source, web-based annotation tool supporting all subtasks of ABSA. AnnoABSA features an intuitive and customizable interface with built-in validation mechanisms and optional LLM-powered suggestions via few-shot prompting. Our evaluation demonstrated that the integrated RAG-based approach, which dynamically leverages the growing pool of annotated examples during the annotation process, significantly outperformed random sampling. A user study revealed significant reductions in annotation time when AI-assisted suggestions are employed.

While this work demonstrates AnnoABSA's functional capabilities and its positive impact on annotation efficiency, future research should systematically investigate how AI assistance affects annotation time, quality, and annotator confidence in ABSA tasks. Such investigations could consider both crowd-sourced workers and domain experts with established ABSA expertise.

Finally, we note that our approach requires no task-specific model training and provides immediate utility upon deployment, making it readily adaptable to diverse NLP annotation scenarios beyond ABSA.

We invite the community to contribute to AnnoABSA's continued development through our GitHub repository. Feature requests in the form of issues and pull requests are welcome, with our commitment to timely integration of valuable contributions to benefit the broader research community.

5. Ethics Statement and Limitations

This research was conducted without industrial funding or commercial sponsorship. We employed AI coding agents Claude Sonnet 4.5[11] and OpenAI's open-source LLM gpt-oss:20b[12] for programming support. Claude Sonnet 4.5 was also used to assist in the formulation of this publication.

Several limitations should be considered when interpreting our results. First, our evaluation of LLM-based suggestions was restricted to Gemma-3-27B due to computational constraints. Larger language models or increased few-shot example sizes could potentially yield superior suggestion quality, albeit at higher computational or financial costs in the case of commercial proprietary LLMs. We executed 580,690 prompts for the evaluation of LLM-based suggestions, which would have incurred substantial costs when employing commercial models. However, AnnoABSA supports arbitrary Ollama and OpenAI-compatible models if one wishes to employ those.

Finally, while we demonstrated significant reductions in annotation time, our evaluation was limited to a small-scale annotation task. The generalizability of these efficiency gains to large-scale real-world scenarios involving thousands of examples

[11] Claude Sonnet: `https://www.anthropic.com/claude/sonnet`

[12] gpt-oss:20b: `https://ollama.com/library/gpt-oss:20b`

remains to be validated, particularly regarding the potential compounding effects of annotator fatigue over extended annotation sessions.

6. Bibliographical References

Samy Ateia and Udo Kruschwitz. 2023. Is chatgpt a biomedical expert? In *Working Notes of the Conference and Labs of the Evaluation Forum (CLEF 2023), Thessaloniki, Greece, September 18th to 21st, 2023*, volume 3497 of *CEUR Workshop Proceedings*, pages 73–90. CEUR-WS.org.

Yoav Benjamini and Yosef Hochberg. 1995. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society: Series B (Methodological)*, 57(1):289–300.

Siva Uday Sampreeth Chebolu, Franck Dernoncourt, Nedim Lipka, and Thamar Solorio. 2024. OATS: A challenge dataset for opinion aspect target sentiment joint detection for aspect-based sentiment analysis. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 12336–12347, Torino, Italia. ELRA and ICCL.

Luigi Colucci Cante, Salvatore D'Angelo, Beniamino Di Martino, and Mariangela Graziano. 2024. *Text Annotation Tools: A Comprehensive Review and Comparative Analysis*, pages 353–362. Springer Nature Switzerland, Cham.

Yong Deng, Xintong Zhang, Danping Zhou, Dequan Zhang, and Boya Huang. 2025. Leveraging nlp in finance: A synergistic approach using large language models and chain-of-thought reasoning. In *Proceedings of the 5th International Conference on Artificial Intelligence and Computer Engineering*, ICAICE '24, page 494–500, New York, NY, USA. Association for Computing Machinery.

Tobias Deußer, Cong Zhao, Daniel Uedelhoven, Lorenz Sparrenberg, Lars Hillebrand, Christian Bauckhage, and Rafet Sifa. 2024. Leveraging large language models for few-shot kpi extraction from financial reports. In *2024 IEEE International Conference on Big Data (BigData)*, pages 4864–4868.

Jakob Fehle, Niklas Donhauser, Udo Kruschwitz, Nils Constantin Hellwig, and Christian Wolff. 2025. German aspect-based sentiment analysis in the wild: B2B dataset creation and cross-domain evaluation. In *Proceedings of the 21st Conference on Natural Language Processing (KONVENS 2025): Long and Short Papers*, pages 213–227, Hannover, Germany. HsH Applied Academics.

Safouane El Ghazouali and Umberto Michelucci. 2025. Visiofirm: Cross-platform ai-assisted annotation tool for computer vision.

Zhibin Gou, Qingyan Guo, and Yujiu Yang. 2023. MvP: Multi-view prompting improves aspect sentiment tuple prediction. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4380–4397, Toronto, Canada. Association for Computational Linguistics.

Md. Arid Hasan, Shudipta Das, Afiyat Anjum, Firoj Alam, Anika Anjum, Avijit Sarker, and Sheak Rashed Haider Noori. 2024. Zero- and few-shot prompting with LLMs: A comparative study with fine-tuned models for Bangla sentiment analysis. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 17808–17818, Torino, Italia. ELRA and ICCL.

Nils Constantin Hellwig, Jakob Fehle, Markus Bink, and Christian Wolff. 2024. GERestaurant: A German dataset of annotated restaurant reviews for aspect-based sentiment analysis. In *Proceedings of the 20th Conference on Natural Language Processing (KONVENS 2024)*, pages 123–133, Vienna, Austria. Association for Computational Linguistics.

Nils Constantin Hellwig, Jakob Fehle, Udo Kruschwitz, and Christian Wolff. 2025. Do we still need human annotators? prompting large language models for aspect sentiment quad prediction. In *Proceedings of the 1st Joint Workshop on Large Language Models and Structure Modeling (XLLM 2025)*, pages 153–172, Vienna, Austria. Association for Computational Linguistics.

Sture Holm. 1979. A simple sequentially rejective multiple test procedure. *Scandinavian journal of statistics*, pages 65–70.

Hannah Kim, Kushan Mitra, Rafael Li Chen, Sajjadur Rahman, and Dan Zhang. 2024. MEGAnno+: A human-LLM collaborative annotation system. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 168–176, St. Julians, Malta. Association for Computational Linguistics.

Jan-Christoph Klie, Michael Bugert, Beto Boullosa, Richard Eckart de Castilho, and Iryna Gurevych.

2018. The INCEpTION platform: Machine-assisted and knowledge-oriented interactive annotation. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, pages 5–9, Santa Fe, New Mexico. Association for Computational Linguistics.

Yanis Labrak, Mickael Rouvier, and Richard Dufour. 2024. A zero-shot and few-shot study of instruction-finetuned large language models applied to clinical and biomedical tasks. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 2049–2066, Torino, Italia. ELRA and ICCL.

Yan Li, Hao Qiu, Xu Wang, Na Dong, and Xinghua Yu. 2025. Rapidx annotator: A specialized software tool for industrial radiographic image annotation and enhancement. *SoftwareX*, 31:102328.

Clément Le Ludec, Maxime Cornet, and Antonio A Casilli. 2023. The problem with annotation. human labour and outsourcing between france and madagascar. *Big Data & Society*, 10(2):20539517231188723.

Yida Mu, Ben P. Wu, William Thorne, Ambrose Robinson, Nikolaos Aletras, Carolina Scarton, Kalina Bontcheva, and Xingyi Song. 2024. Navigating prompt complexity for zero-shot classification: A study of large language models in computational social science. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 12074–12086, Torino, Italia. ELRA and ICCL.

Arbi Haza Nasution and Aytuğ Onan. 2024. Chatgpt label: Comparing the quality of human-generated and llm-generated annotations in low-resource language nlp tasks. *IEEE Access*, 12:71876–71900.

Gaurav Negi, Rajdeep Sarkar, Omnia Zayed, and Paul Buitelaar. 2024. A hybrid approach to aspect based sentiment analysis using transfer learning. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 647–658, Torino, Italia. ELRA and ICCL.

Jingwei Ni, Minjing Shi, Dominik Stammbach, Mrinmaya Sachan, Elliott Ash, and Markus Leippold. 2024. AFaCTA: Assisting the annotation of factual claim detection with reliable LLM annotators. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1890–1912, Bangkok, Thailand. Association for Computational Linguistics.

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammad AL-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, Véronique Hoste, Marianna Apidianaki, Xavier Tannier, Natalia Loukachevitch, Evgeniy Kotelnikov, Nuria Bel, Salud María Jiménez-Zafra, and Gülşen Eryiğit. 2016. SemEval-2016 task 5: Aspect based sentiment analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 19–30, San Diego, California. Association for Computational Linguistics.

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Suresh Manandhar, and Ion Androutsopoulos. 2015. SemEval-2015 task 12: Aspect based sentiment analysis. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 486–495, Denver, Colorado. Association for Computational Linguistics.

Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. SemEval-2014 task 4: Aspect based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 27–35, Dublin, Ireland. Association for Computational Linguistics.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389.

Ariana Sahitaj, Premtim Sahitaj, Veronika Solopova, Jiaao Li, Sebastian Möller, and Vera Schmitt. 2025. Hybrid annotation for propaganda detection: Integrating LLM pre-annotations with human intelligence. In *Proceedings of the Fourth Workshop on NLP for Positive Impact (NLP4PI)*, pages 215–228, Vienna, Austria. Association for Computational Linguistics.

Somayeh Mehrizi Sani and Yeganeh Keyvan Shokooh. 2016. Minimalism in designing user interface of commercial websites based on gestalt visual perception laws (case study of three top brands in technology scope). In *2016 Second International Conference on Web Research (ICWR)*, pages 115–124.

Hope Schroeder, Deb Roy, and Jad Kabbara. 2025. Just put a human in the loop? investigating LLM-assisted annotation for subjective tasks. In *Findings of the Association for Computational*

Linguistics: ACL 2025, pages 25771–25795, Vienna, Austria. Association for Computational Linguistics.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. 2025. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*.

An Wang, Junfeng Jiang, Youmi Ma, Ao Liu, and Naoaki Okazaki. 2023. Generative data augmentation for aspect sentiment quad prediction. pages 128–140.

Michael Wojatzki, Eugen Ruppert, Sarah Holschneider, Torsten Zesch, and Chris Biemann. 2017. Germeval 2017: Shared task on aspect-based sentiment in social media customer feedback. *Proceedings of the GermEval*, pages 1–12.

Hongling Xu, Yice Zhang, Qianlong Wang, and Ruifeng Xu. 2025. DS^2-ABSA: Dual-stream data synthesis with label refinement for few-shot aspect-based sentiment analysis. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15460–15478, Vienna, Austria. Association for Computational Linguistics.

Wenxuan Zhang, Yang Deng, Xin Li, Yifei Yuan, Lidong Bing, and Wai Lam. 2021. Aspect sentiment quad prediction as paraphrase generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9209–9219, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Wenxuan Zhang, Yue Deng, Bing Liu, Sinno Pan, and Lidong Bing. 2024. Sentiment analysis in the era of large language models: A reality check. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 3881–3906, Mexico City, Mexico. Association for Computational Linguistics.

Wenxuan Zhang, Xin Li, Yang Deng, Lidong Bing, and Wai Lam. 2023. A Survey on Aspect-Based Sentiment Analysis: Tasks, Methods, and Challenges . *IEEE Transactions on Knowledge & Data Engineering*, 35(11):11019–11038.

Changzhi Zhou, Dandan Song, Yuhang Tian, Zhijing Wu, Hao Wang, Xinyu Zhang, Jun Yang, Ziyi Yang, and Shuhao Zhang. 2024. A comprehensive evaluation of large language models on aspect-based sentiment analysis. *arXiv preprint arXiv:2412.02279*.

A. ABSA Tasks

Task Type	Task Name	Elements	Gold Label (Example)
Single	Aspect Term Extraction (ATE)	*a*	`['waiter']`
	Aspect Category Detection (ACD)	*c*	`['food quality', 'service speed']`
	Opinion Term Extraction (OTE)	*o*	`['delicious', 'way too slow']`
Compound	Aspect Sentiment Classification (ASC)	*a*, *s*	`[('waiter', 'negative')]`
	Aspect-Opinion Pair Extraction (AOPE)	*a*, *o*	`[('waiter', 'way too slow')]`
	End-to-End ABSA (E2E-ABSA)	*a*, *s*	`[('NULL', 'positive'), ('waiter', 'negative')]`
	Aspect Category Sentiment Analysis (ACSA)	*c*, *s*	`[('food quality', 'positive'), ('service speed', 'negative')]`
	Aspect Sentiment Triplet Extraction (ASTE)	*a*, *o*, *s*	`[('NULL', 'delicious', 'positive'), ('waiter', 'way too slow', 'negative')]`
	Target Aspect Sentiment Detection (TASD)	*a*, *c*, *s*	`[('NULL', 'food quality', 'positive'), ('waiter', 'service speed', 'negative')]`
	Aspect Sentiment Quad Prediction (ASQP)	*a*, *c*, *o*, *s*	`[('NULL', 'food quality', 'delicious', 'positive'), ('waiter', 'service speed', 'way too slow', 'negative')]`
	Aspect-Category-Opinion-Sentiment Quad Extraction (ACOS)	*a*, *c*, *o*, *s*	`[('NULL', 'food quality', 'delicious', 'positive'), ('waiter', 'service speed', 'way too slow', 'negative')]`

Table 3: **Overview of ABSA tasks supported by AnnoABSA.** For each task, a gold label is presented for the sentence *"It was really delicious, but the waiter was way too slow"*. Aspect categories are commonly selected from a predefined set ({food quality, service speed, ...}). Notation: *a* = aspect term, *c* = aspect category, *o* = opinion term, *s* = sentiment polarity. In case of an implicit aspect, aspect term *a* is set to `'NULL'`. The listed tasks are equivalent to those reported in the literature review by Zhang et al. (2023)

B. CLI Flags

Option	Description	Default
Server Configuration		
`–backend`	Start only backend server	–
`–backend-port`	Backend server port	`8000`
`–frontend-port`	Frontend server port	`3000`
`–backend-ip`	Backend server IP address	`localhost`
`–frontend-ip`	Frontend server IP address	`localhost`
Session Management		
`–session-id`	Unique identifier for an annotation session.	`None`
Annotation Elements		
`–elements`	Sentiment elements to annotate	aspect_term, aspect_category, sentiment_polarity, opinion_term
`–polarities`	Valid sentiment polarities	positive, negative, neutral
`–categories`	Valid aspect categories	Restaurant domain (13 categories)
`–implicit-aspect`	Allow implicit aspect terms	Enabled
`–disable-implicit-aspect`	Disable implicit aspect terms	Disabled
`–implicit-opinion`	Allow implicit opinion terms	Disabled
`–disable_implicit_opinion`	Disable implicit opinion terms	Enabled
Interface and Processing		
`–disable_clean_phrases`	Disable automatic punctuation cleaning from phrase start/end	Enabled
`–disable-save-positions`	Disable saving phrase positions	Enabled
`–disable-click-on-token`	Disable click-on-token feature	Enabled
`–auto-positions`	Enable automatic position filling on startup	Disabled
`–annotation-guidelines`	Path to PDF file containing annotation guidelines	Disabled
Analytics and Timing		
`–store-time`	Store timing data for annotation sessions	Disabled
`–display-avg-annotation-time`	Display average annotation time in the interface	Disabled
AI Integration		
`–ai-suggestions`	Enable AI-powered prediction suggestions using LLM	Disabled
`–disable-ai-automatic-prediction`	Disable automatic AI prediction triggering	Disabled
`–llm-model`	LLM employed for suggestions (e.g., `gemma3:4b`, `gpt-4o`)	`gemma3:4b`
`–openai-key`	OpenAI API key for using OpenAI models	–
`–n-few-shot`	Maximum number of few-shot examples in LLM context	`10`
Configuration Management		
`–save-config`	Save config to JSON file	–
`–show-config`	Display current configuration in console	–

Table 4: **CLI options for AnnoABSA.** AnnoABSA offers extensive customization across server settings, annotation logic, and AI integration.

C. Prompt for RAG-Based Suggestions

```
According to the following sentiment elements definition:

- The 'aspect term' is the exact word or phrase in the text that represents a specific feature, attribute, or aspect of a product or service that a user
may express an opinion about. The aspect term might be 'NULL' for implicit aspect.
- The 'aspect category' refers to the category that aspect belongs to, and the available categories includes: hotel comfort, rooms design_features,
facilities miscellaneous, rooms cleanliness, food_drinks miscellaneous, food_drinks style_options, facilities comfort, room_amenities general, service
general, rooms quality, rooms general, hotel general, food_drinks prices, facilities general, room_amenities quality, facilities quality, rooms
miscellaneous, facilities design_features, hotel prices, food_drinks quality, room_amenities prices, room_amenities comfort, rooms prices, hotel
cleanliness, hotel miscellaneous, facilities prices, rooms comfort, hotel quality, room_amenities design_features, location general, facilities
cleanliness, hotel design_features, room_amenities cleanliness.
- The 'sentiment polarity' refers to the degree of positivity, negativity or neutrality expressed in the opinion towards a particular aspect or feature of
a product or service, and the available polarities include: neutral, negative, positive.
- The 'opinion term' is the exact word or phrase in the text that refers to the sentiment or attitude expressed by a user towards a particular aspect or
feature of a product or service.

Recognize all sentiment elements with their corresponding aspect terms, aspect categorys, sentiment polaritys, opinion terms in the following text in the
form of a list of objects, each object having key(s) 'aspect term', 'aspect category', 'sentiment polarity', 'opinion term'.

Here are some examples:
Text: This is a budget hotel in a good location central to everything .
Sentiment elements: [('aspect term': 'hotel', 'aspect category': 'hotel prices', 'sentiment polarity': 'positive', 'opinion term': 'budget'), ('aspect
term': 'hotel', 'aspect category': 'location general', 'sentiment polarity': 'positive', 'opinion term': 'good location')]
Text: it was very relaxing good buffet breakfast included and so close to everything .
Sentiment elements: [('aspect term': 'breakfast', 'aspect category': 'food_drinks quality', 'sentiment polarity': 'positive', 'opinion term': 'relaxing'),
('aspect term': 'breakfast', 'aspect category': 'food_drinks quality', 'sentiment polarity': 'positive', 'opinion term': 'good')]
Text: Loved our Spaulding Hotel Very good location , centrally located close to Theaters and shopping .
Sentiment elements: [('aspect term': 'Spaulding Hotel', 'aspect category': 'hotel general', 'sentiment polarity': 'positive', 'opinion term': 'Loved'),
('aspect term': 'location', 'aspect category': 'location general', 'sentiment polarity': 'positive', 'opinion term': 'Very good'), ('aspect term':
'Spaulding Hotel', 'aspect category': 'location general', 'sentiment polarity': 'positive', 'opinion term': 'centrally located close to Theaters and
shopping')]
Text: Good restaurants nearby (
Sentiment elements: [('aspect term': 'NULL', 'aspect category': 'location general', 'sentiment polarity': 'positive', 'opinion term': 'Good restaurants
nearby')]
Text: well thought out design to incorporate alot of extras .
Sentiment elements: [('aspect term': 'NULL', 'aspect category': 'rooms design_features', 'sentiment polarity': 'positive', 'opinion term': 'well thought
out design')]
Text: There are a number of restaurants in walking distance .
Sentiment elements: [('aspect term': 'NULL', 'aspect category': 'location general', 'sentiment polarity': 'positive', 'opinion term': 'number of
restaurants in walking distance')]
Text: The owners and staff are extremely accomodating , offering good advice to restaurants
Sentiment elements: [('aspect term': 'owners', 'aspect category': 'location general', 'sentiment polarity': 'positive', 'opinion term': 'extremely
accomodating'), ('aspect term': 'staff', 'aspect category': 'location general', 'sentiment polarity': 'positive', 'opinion term': 'offering good advice to
restaurants')]
Text: Outside is a magnificent view of the river for which the hotel is named and parallel is a cobblestone street filled with a wide range of charming
shops .
Sentiment elements: [('aspect term': 'hotel', 'aspect category': 'hotel design_features', 'sentiment polarity': 'positive', 'opinion term': 'magnificent
view of the river')]
Text: The only thing that the hotel could improve on was seating in the main lobby -  - there is a couple of chairs and a couch and alot of empty space .
Sentiment elements: [('aspect term': 'hotel', 'aspect category': 'facilities comfort', 'sentiment polarity': 'neutral', 'opinion term': 'improve on was
seating in the main lobby')]
Text: great location close to amenities , but nice and quiet on an evening .
Sentiment elements: [('aspect term': 'location', 'aspect category': 'location general', 'sentiment polarity': 'positive', 'opinion term': 'great')]
Text: It is in a good location and close to alot of shops and restaurants .
Sentiment elements:
```

Figure 4: **Prompt used for RAG-based suggestion prediction.** The prompt includes a task description with explanations of sentiment elements, ten in-context demonstrations, and the target text for aspect prediction. The few-shot examples shown are taken from the Hotels dataset and include annotations for ASQP