LLM-Agent Support for Two-Document Comparison Using Hierarchical Topic Maps

Mariia Tytarenko* Graz University Of Technology

Tobias Walter Rutar† Graz University Of Technology

Stefan Lengauer‡ Graz University Of Technology

Tobias Schreck§ Graz University Of Technology

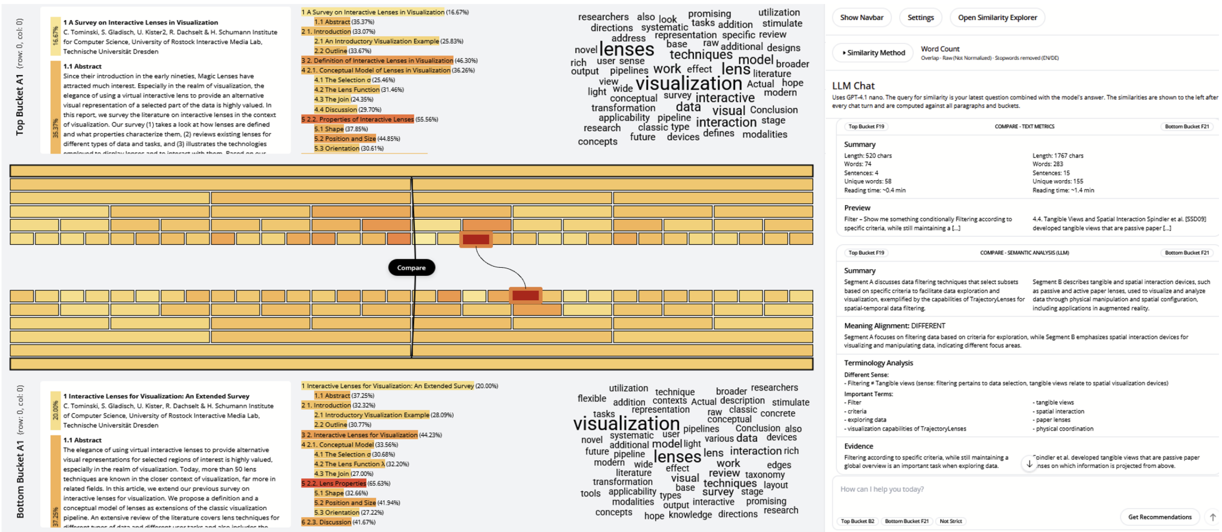

Figure 1: The bipartite interface for the document comparison, as it is displayed to the user. Left: two documents are displayed as synchronized HTMs with a colorization based on the selected keywords. Right: Parameter selection and Chatbot interface.

Abstract

Comparing two related documents to uncover subtle similarities, differences, and shifts in emphasis is an intellectually demanding task. In a previous work, we proposed an organized outline to compare two documents in a so-called HTM, providing a per-segment comparison which highlights sematic similarities across the documents through an icicle-plot-like layout. We found that users still face the challenge of unifying terminology, clarifying meanings across diverging wordings, and thus drawing conclusions regarding the actual overlap or differences between two paragraphs. To address this shortcoming, we extend the HTM by equipping it with an Large Language Model (LLM) agent. Our agent aims to support – the otherwise purely manual document comparison process – giving concise, evidence-based summaries. Specifically, it highlights whether passages share the same meaning or take up different standpoints, it reveals synonyms and term drift, it answers targeted questions and it recommends the most similar next segments to discover, based on users' exploration provenance. Focusing on document comparison, we present two scenario-based use cases and discuss implications for designing agents that support users by guiding them towards more grounded conclusions.

Index Terms: Text visualization, sensemaking, large language models, agents

*e-mail: mariia.tytarenko@tugraz.at

†e-mail: tobias.rutar@student.tugraz.at

‡e-mail: s.lengauer@tugraz.at

§e-mail: tobias.schreck@tugraz.at

1 Introduction

Text pervades both everyday life and professional practice, ranging from news and research reports to clinical guidelines and patient brochures. The sheer volume generated texts and their complexity constitute both exciting opportunities and substantial challenges for anyone trying to analyze it. Clinicians, researchers, and guideline developers often need to compare two documents (whether it is about understanding differences between versions of a same document or about revising editions of an information booklet) and have to produce conclusions that are not just insightful, but also verifiable. The field of Visual Analytics (VA) addresses this issue by coupling computational analysis with interactive views, in order to externalize reasoning and reduce cognitive load. Still, when document curators are tasked to compare two specific documents and decide whether to replace one of them, they face a series of demanding challenges. Informed by discussions with experts, we identified four core challenges: (1) *Targeted selection* – users often need help in directing attention to the most similar or most dissimilar segments (paragraphs/sections) without completely reading both texts; (2) *Semantic reconciliation* – documents may express the same idea with different manifestations (analysts are contrasting concepts, not strings); (3) *Context and provenance* – simple color coding of document differences is a common practice and guides users *where* to look, but it does not reveal *why* (quoted sentences and traceable evidence is required to backup decisions); **Guided progression** – users benefit in guidance on how to proceed through the comparison in a sensible order with suggestions for what to examine next.

LLMs show promising capabilities as intelligent agents in open-context question answering as they excel in interpreting users' intent and following instructions [3]. Visualization researchers employ LLMs to assist with data conversion [22] and even generate visualizations [15, 11, 14].

Building on previous research in comparative text visualization, HTM [21] is a side-by-side interface that color-codes the similarity of document segments to a user-defined keyword query. It uses cell-wise similarity to effectively measures the density of query terms, helping users to identify overlaps and differences in a document's structure. While this score is efficient and stable, it remains lexical and cannot by itself explain similarity at a semantic level (e.g., synonyms, paraphrases, rephrasings). In this paper we present an addition to the HTM, in the form of an LLM-based agent, next to the base visualization. If a user selects one or two cells in the HTM the agent acts as a reasoning partner. I.e., it (i) explains why the text cells were highlighted as being similar or different, (ii) resolves terminology (handling synonymy), (iii) marks contradictions or gaps, (iv) answers user questions, and (v) proposes intuitive next steps in the form of similar segments to discover. In summary, the HTM indicates *where to look*; the agent explains *what it means* and proposes *what to do next*.

2 Related Work

Comparative text visualization has a rich history. For the efficient consumption of large text corpora, it is beneficial to view them with some degree of abstraction, using custom-built techniques. This is referred to as *distant reading* [16] and constitutes the opposite of the natural *close reading* – the "conventional" linear reading fashion of word by word from top to bottom. Early works, such as TileBar visualization [9] split documents and highlight segments by query term frequency, giving a visual understanding of where useful information resides. In a similar fashion, *literature fingerprinting* [12] is used to reveal the variability of textual features over the linear structure of a text. In addition, the work by Ben Fry [6] demonstrates how tracing textual evolution between versions can detect subtle evolution, inspiring to focus on meaningful, evidence-based comparison. There are also approaches providing a holistic overview on a document's textual and pictorial content, such as *document cards* [17] which convey the essence of a document in a minimalist space-saving manner. Our previous approach HTM [21] organizes a document into a hierarchy of document segments. It encodes topic distributions across levels, enabling drill-down and cross-document thematic comparisons. As opposed to topic-based views abstracting text as sequences of word-based topics [5, 8, 13], embedding-based approaches explicitly represent a segment/document as a dense vector [7, 1], aligning the unit of abstraction with the analyst's unit of analysis (segment/document).

More recently LLMs experienced increasing employment as agentic helpers for solving VA tasks. Hypergraph Visualization and INTelligent Agentss (HINTs) [14] illustrate a large corpus of documents as a hypergraph, groups it, and uses an LLM agent both to detect salient entities and to enable open-ended analysis through a chatbot. LLM-Enhanced VA (VA) [26] employs LLMs for different workflows – i.e., onboarding, exploration (suggestion of findings), and summarization – showing how it can help analysis and report generation. WaitGPT [23] offers a way to visualize code generated by LLMs for better transparency. LightVA [25] introduces a planner-executor-controller system that breaks down goals, conducts analyses, and creates visualizations. Hierarchical and relational views such as HTM and HINT reveal structure but leave semantic relationships open to be interpreted by the user.

The strength of our proposed system is that it neatly combines a conventional approach for document comparison (in the form of HTM) with an LLM-agent. The benefit of the latter is that it allows for much more natural interaction with the tool through natural language and more fine-grained notion on text similarity or dissimilarity as opposed to engineered features.

3 System Overview

Our system is designed for experts and analysts who compare and make sense of a pair of similar documents. While basically any pair of documents can be loaded, the HTM is particularly useful for analyzing different versions or editions of one-and-the-same document. Such a comparison task is a common objective when curating a library/knowledge repository. The core question is whether a new version of a document contains sufficient changes to merit an update. To this end, a deep understanding of (subtle) differences is needed. Other possible use cases include the comparison of different translations of documents on a semantic level.

Building on our previous work, we place two HTMs (using a common set of keywords) facing each other to reveal differences in the documents' structure (Fig. 1). Additionally, we draw linking lines between cells with an above-threshold similarity and add an LLM agent that explains the their similarities or differences. Besides that, it guides a user on what to do next. The visualization serves to orient – where to look and to understand a structure – and the agent to supply semantics and why it matters. We support multiple ways for achieving a meaningful selection of keywords: (i) a user can manually type-in words; (ii) select from the most prevalent terms displayed in a Word Cloud [10]; (iii) select a set from a given list of semantically similar concepts – i.e., topics which are derived from the documents' using a topic modeling approach [4]. For the latter, we use Latent Dirichlet Allocation [2] for a single-document and Hierarchic Dirichlet Process (HDP) [18] for a document comparison. The method for similarity computation can be selected by the user. Currently, our system supports multiple options such as Word Count (i.e., literal overlap), TF/TF-IDF, latent distributional similarity, or semantic embeddings, which are robust to synonyms.

We intentionally separate the keyword-based orientation aid (the HTM) from the embedding-based semantic guidance (the agent) to ensure clarity and maintain a clear distinction between visual cues and model interpretations. In the background, the system stores the session context (selected cells, comparison outputs, keywords and chat history) and uses a LLM to generate a short session summary. Prior work showed that LLMs qualify as intelligent agents for sensemaking [14]. We carry this idea over to our document comparison: when the user selects a highlighted cell pair, the agent retrieves the corresponding texts and metadata and invokes a LLM to produce an explanation and suggestion, based on a predefined prompt. Design goals in our approach are: (G1) keep explanations within user-selected cells; (G2) make terms reconciliation explicit; (G3) offer different similarity methods and (G4) provide actionable next actions based on cell similarity.

The outcome is a well-organized workflow: the HTMs highlight where important information can be found through color-coding and visual links, while the agent clarifies how different cells relate to each other and answers focused questions and offers suggestions on what to explore next.

4 Agent Design

Our system uses a two-stage agent for LLM-invocation. Stage 1 produces a grounded comparison of the currently-selected cells, while Stage 2 proposes a new high-value topic and identifies specific locations to read next. Both stages operate solely on output from the visualization pipeline and current session state; neither modifies the HTMs nor accesses external resources.

Our agent employs a minimalist Brain–Memory–Tools [24] architecture. Each stage consists of a single LLM invocation, parameterized by a fixed prompt template that specifies inputs, guardrails, and a strict JSON output schema. The prompt is treated as the Brain's executable specification. Memory retains only session-critical context: the validated keyword query, recent comparison results, chat excerpts, the selected similarity method, and lists of items already presented to the user (e.g., shown topics).

Stage 1 calls Comparison Prompt (Listing 1) with `Segment A` from (the top HTM) and `Segment B` (the bottom HTM). The agent replies with strict JSON containing: short summaries of `A`/`B`; a meaning alignment reasoning (`SAME` or `DIFFERENT`); a terminology analysis separating synonyms/paraphrases from same-term–different-sense and only-in-`A`/`B` terms; and one short quote from each segment as provenance.

Listing 1: The comparison prompt for Stage 1.

```
COMPARE_PROMPT = """
You are a document-comparison assistant. Use ONLY the provided text segments.

Task
Compare two text segments (A and B). Summarize them, determine whether they express the
        same or different meanings, analyze terminology overlap, and extract evidence.

Guidelines
- Summarize A and B in 1-2 sentences each.
- Declare whether A and B express the SAME concept or DIFFERENT meanings; give a short
        rationale.
- Terminology analysis:
  * List synonyms/paraphrases as ‘A_term <--> B_term‘.
  * Mark same terms used with different sense as ‘term != term (sense: ...)‘.
  * Note important terms that appear only in A or only in B.
- Evidence:
  * Quote one short sentence from A.
  * Quote one short sentence from B.
- Be concise and structured.
- If evidence is insufficient, say so.

Output strictly as JSON in this schema:
{{
  "summary": {{
    "A": "<1-2 sentences summarizing A>",
    "B": "<1-2 sentences summarizing B>"
  }},
  "meaning_alignment": {{
    "relation": "SAME | DIFFERENT",
    "rationale": "<short explanation>"
  }},
  "terminology": {{
    "synonyms": ["A_term <--> B_term", ...],
    "different_sense": ["term != term (sense: ...)", ...],
    "only_in_A": ["term1", "term2", ...],
    "only_in_B": ["term1", "term2", ...]
  }},
  "evidence": {{
    "A": "<short quote from A>",
    "B": "<short quote from B>"
  }}
}}

CONTEXT
Segment A: <<< {A_TEXT} >>>
Segment B: <<< {B_TEXT} >>> """
```

When the user requests suggestions, we run a second prompt (Stage 2) in order to recommend most relevant next segment to explore. The Recommendation prompt (Listing 2) takes two inputs: the aggregated session context (recent selected cells, chat history, comparison outputs) and the set of topics already presented. The agent must output exactly one topic with a title, description (1-2 sentences), and reasoning (1-2 sentences) explaining its relevance to the current session. The system then select the most similar paragraph(s) and suggests the corresponding top unexplored cells in a form of clickable buttons.

Listing 2: The recommendation prompt for Stage 2.

```
RECOMMENDATION_GROUP_PROMPT = """
You are a recommendation generator. Propose **one concise, high-value topic** based on the
        given group of paragraphs.

Language
- Use the SAME LANGUAGE as the session context. If context is German, respond in German; if
         English, respond in English.

Task
- Read the session context and the current focus buckets (A/B) for situational relevance.
- Read ONLY the provided paragraph group text.
- Propose exactly ONE topic that would be useful to explore next.
- Fields:
  * "title": short and catchy {(26 words)
  * "description": {12 sentences, what it covers
  * "reasoning": {12 sentences why ’its relevant now, ideally referencing the context

Output strictly as JSON:
{{
  "topic": {{
    "title": "<short title>",
    "description": "<1-2 sentences>",
    "reasoning": "<1-2 sentences>"
  }}
}}

SESSION CONTEXT <<< {context} >>>
CURRENT FOCUS BUCKETS (A/B) <<< {focus_buckets} >>>
PARAGRAPH GROUP TEXT <<< {group_text} >>>
```

5 Use Cases

Through two carefully-selected use cases, we demonstrate how our system can be used to compare two similar documents with the help of an LLM agent.

For the first use case, we compare two related survey papers on interactive lenses in visualization by Tominski et al. [19, 20], as we did with the original HTM approach [21]. With the help of our agent, a user is guided in understanding how the surveys relate and where they differ, to unveil recent developments and to better understand the documents.

The process is initiated by uploading the two documents and choosing an initial keyword query: the user chooses topics obtained through the HDP, i.e., $\{spatial, automatically, volume, flow, sampling\}$. This keyword query is applied to colorize both HTMs (Fig. 2). The visual link between the two most similar cells across the documents features a "Compare" button. If this button is clicked, the system initiates the comparison agent (Stage 1). The agent presents a formatted explanation in the chat on the right-hand side. Regarding the selected cells the agent reports:

> Segment A focuses on filtering data based on criteria for exploration, while Segment B emphasizes spatial interaction devices for visualizing and manipulating data, indicating different focus areas.

That is, they differ by scope/focus. This one line has explicit analytic traction. If the analyst intends to harmonize or merge the documents, it reveals an editorial gap: the filtering-centric discussion (A) does not address device-mediated interactions, while the device-centric discussion (B) does not ground filtering workflows. After aligning the terminology, the analyst's next step is to see if the contrasting ideas are explored further in the documents. To this end, the agent summarizes the main focus of the session, creates a clear topic that captures the discussion, and then returns matching paragraphs that are linked to respective cells through clickable buttons. Relying solely on the HTM might lead the analyst to the most vividly colored neighboring segments. However, in our use case, the agent takes into account the session history and then employs paragraph-level embeddings to rank what is most semantically relevant to the current conversation. As a result, it might recommend diving into a less prominently colored segment that better continues the identified thread.

For the second use case we compared two versions of the *Type 2 Diabetes* article from Wikipedia[1] (from 2023 and 2025), selecting the appropriate keywords $\{diagnosis, glycated\ hemoglobin, HbA1c, screening\}$. The system output (Fig. 3) clearly shows that the core diagnostic criteria remain unchanged, but highlights missing information in the previous version, such as the use of glycated hemoglobin (HbA1c) as a measure during diagnosis and revised screening guidelines. This makes it simple for a curator to easily identify what is outdated and make an evident decision in including the new information.

This highlights the strength of our design: by separating keyword-based orientation (coloring and linking) from embedding-based guidance (recommendations), the system transforms visual cues into solid explanations and actionable next steps, even when the vocabulary shifts. The resulting workflow facilitates sharper comparisons, more focused discussions, and quicker decisions on what to read (or edit) next across the two documents.

6 Future Work and Discussion

Our prototype demonstrates a disciplined split – keywords for orientation, embeddings for guidance – but also comes with respective advantages and limitations. In addition to scenario walkthrough,

[1] https://en.wikipedia.org/wiki/Type_2_diabetes

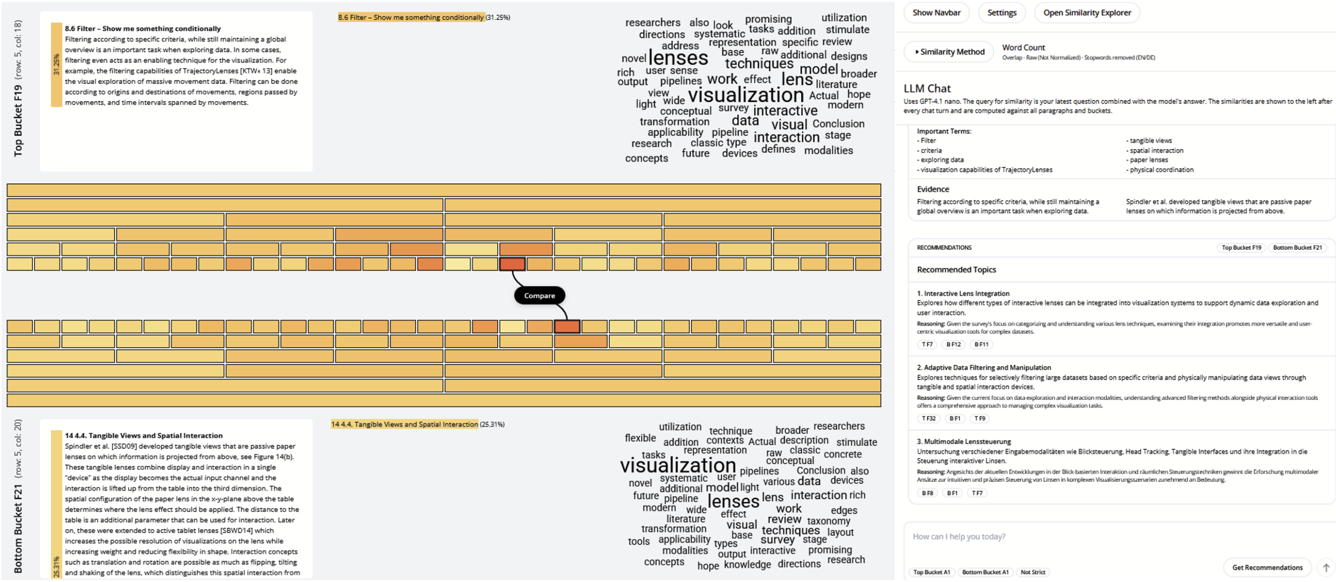

Figure 2: Left: connected segments across the two documents. Right: chatbot-generated recommendations.

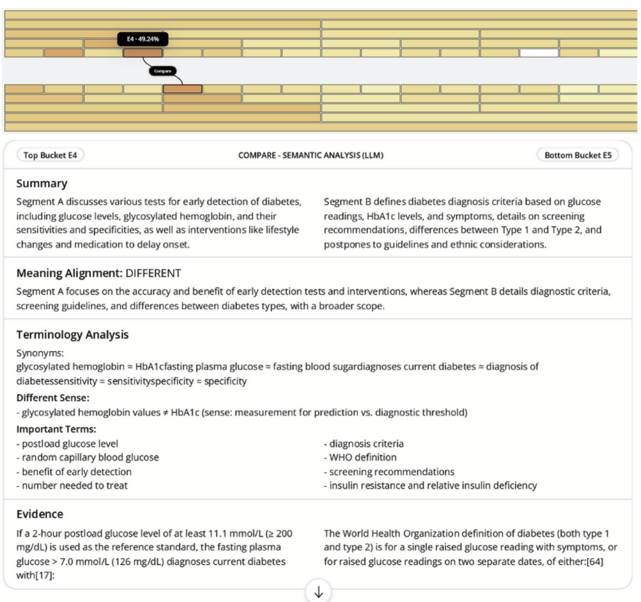

Figure 3: System output comparing two article segments. Shows a difference between segments: one emphasizes early detection tests, while the other defines diagnostic and screening recommendations.

there is a need for controlled user studies measuring: the accuracy of meaning-alignment (judged against expert annotations); provenance quality (are the cited spans sufficient and correct?); efficiency; effectiveness of next-segment recommendations; experienced cognitive load; and trust.

Our agent is currently working with a very limited session memory. The next step involves creating a structured session state that captures open questions, unresolved differences, accepted alignments, and information of what the analyst found useful. This would allow for more goal-oriented reasoning – addressing the "scope mismatch" – and ensure that any continuations take previous decisions into account. To improve our system, there is a need for more autonomous agent that can actively suggest additional keywords, switch similarity modes when needed, and initiate the next steps or lightweight visualization actions to keep the analysis flowing. An agent could (i) propose additional keywords when the current query is weekly represented in the documents; (ii) proactively surface bridging cells after detecting a difference; and (iii) issue micro-prompts to verify whether a recommendation actually resolved an ambiguity. The agent could reason on how a concept develops through different sections (introduction – specialization – operationalization), with citations. Such synopses would directly support harmonizing two versions of a document or drafting a merged narrative.

7 Conclusion

This paper outlines a two-document comparison system with an LLM-based agent and a keyword-curated, HTM-based visual substrate. Extending our HTM approach, we presented a concept how a LLM can be integrated as an agent, supporting the user in document comparison and – particularly – in understanding the subtle differences and similarities. We believe, that the VA task of document comparison merits additional research in the direction of using document visualizations in conjunction with LLMs.

Our approach makes the user intent explicit through selected keywords and reveals relevant text passages togehter with their appearance context with color-coded hierarchies. By selecting follow-up segments from system-generated similarity lists – and by recommending additional cells – the agent maintains traceability while delivering purposeful suggestions. In summary, we present a well-designed, reproducible process for comparative reading in high-reliability areas such as research papers, policy texts etc. There are ample possibilities for future extensions, such as mixed-initiative keyword refinement, different recommendation options, bias detection, well-contained agent autonomy as well as visualization awareness, and the possibility to directly influence it.

Acknowledgments

This work was funded by the Austrian Science Fund (FWF) as part of the project 'Human-Centered Interactive Adaptive Visual Approaches in High-Quality Health Information' (Adaptive and interactive CHIS (A$^+$CHIS); Grant No. FG 11-B).

References

[1] D. Angelov. Top2vec: Distributed representations of topics, 2020. 2

[2] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3(null):993–1022, mar 2003. 2

[3] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners. *CoRR*, abs/2005.14165, 2020. 1

[4] W. Ding and C. Chen. Dynamic topic detection and tracking: A comparison of HDP, C-word, and cocitation methods. *Journal of the Association for Information Science and Technology*, 65(10):2084–2097, 2014. doi: 10.1002/asi.23134 2

[5] W. Dou, L. Yu, X. Wang, Z. Ma, and W. Ribarsky. Hierarchicaltopics: Visually exploring large text collections using topic hierarchies. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2002–2011, 2013. doi: 10.1109/TVCG.2013.162 2

[6] B. Fry. On the origin of species: The preservation of favoured traces. *Ben Fry*, 2009. 2

[7] M. Grootendorst. Bertopic: Neural topic modeling with a class-based tf-idf procedure, 2022. 2

[8] D. Han, G. Parsad, H. Kim, J. Shim, O.-S. Kwon, K. A. Son, J. Lee, I. Cho, and S. Ko. Hisva: A visual analytics system for studying history. *IEEE Transactions on Visualization and Computer Graphics*, 28(12):4344–4359, 2022. doi: 10.1109/TVCG.2021.3086414 2

[9] M. A. Hearst. TileBars: visualization of term distribution information in full text information access. In *International Conference on Human Factors in Computing Systems*, 1995. 2

[10] F. Heimerl, S. Lohmann, S. Lange, and T. Ertl. Word Cloud Explorer: Text Analytics Based on Word Clouds. In *2014 47th Hawaii International Conference on System Sciences*, pp. 1833–1842, 2014. doi: 10.1109/HICSS.2014.231 2

[11] A. Jobst, D. Atzberger, W. Scheibel, J. Döllner, and T. Schreck. Delphi: A natural language interface for 2.5d treemap visualization of source code. In *Proceedings of the 20th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications – IVAPP*, IVAPP '25, pp. 867–874. INSTICC, SciTePress, 2025. doi: 10.5220/0013119600003912 1

[12] D. A. Keim and D. Oelke. Literature fingerprinting: A new method for visual literary analysis. In *2007 IEEE Symposium on Visual Analytics Science and Technology*, pp. 115–122. IEEE, 2007. 2

[13] H. Lee, J. Kihm, J. Choo, J. Stasko, and H. Park. ivisclustering: An interactive visual document clustering via topic modeling. *Computer Graphics Forum*, 31(3pt3):1155–1164, 2012. doi: 10.1111/j.1467-8659.2012.03108.x 2

[14] S. Y.-T. Lee and K.-L. Ma. HINTs: Sensemaking on Large Collections of Documents With Hypergraph Visualization and INTelligent Agents. *IEEE Transactions on Visualization and Computer Graphics*, 31(9):5532–5546, Sept. 2025. doi: 10.1109/tvcg.2024.3459961 1, 2

[15] P. Maddigan and T. Susnjak. Chat2vis: Generating data visualisations via natural language using chatgpt, codex and gpt-3 large language models, 2023. 1

[16] F. Moretti. *Graphs, Maps, Trees: Abstract Models for a Literary History*. Verso, 2005. 2

[17] H. Strobelt, D. Oelke, C. Rohrdantz, A. Stoffel, D. A. Keim, and O. Deussen. Document cards: A top trumps visualization for documents. *IEEE transactions on visualization and computer graphics*, 15(6):1145–1152, 2009. 2

[18] Y. Teh, M. Jordan, M. Beal, and D. Blei. Hierarchical Dirichlet Processes. *Machine Learning*, pp. 1–30, 12 2006. doi: 10.1198/016214506000000302 2

[19] C. Tominski, S. Gladisch, U. Kister, R. Dachselt, and H. Schumann. A Survey on Interactive Lenses in Visualization. In *EuroVis - STARs*. The Eurographics Association, 2014. doi: 10.2312/eurovisstar.20141172 3

[20] C. Tominski, S. Gladisch, U. Kister, R. Dachselt, and H. Schumann. Interactive Lenses for Visualization: An Extended Survey. *Computer Graphics Forum*, 36(6):173–200, 2017. doi: 10.1111/cgf.12871 3

[21] M. Tytarenko, L. Shao, T. W. Rutar, M. A. Bedek, C. Krenn, S. Lengauer, and T. Schreck. Hierarchical Topic Maps for Visual Exploration and Comparison of Documents . In *EuroVis Workshop on Visual Analytics (EuroVA)*. The Eurographics Association, 2024. doi: 10.2312/eurova.20241119 2, 3

[22] C. Wang, J. Thompson, and B. Lee. Data formulator: Ai-powered concept-driven visualization authoring, 2023. 1

[23] L. Xie, C. Zheng, H. Xia, H. Qu, and C. Zhu-Tian. Waitgpt: Monitoring and steering conversational llm agent in data analysis with on-the-fly code visualization. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*, UIST '24, p. 1–14. ACM, Oct. 2024. doi: 10.1145/3654777.3676374 2

[24] M. Yu, F. Meng, X. Zhou, S. Wang, J. Mao, L. Pang, T. Chen, K. Wang, X. Li, Y. Zhang, B. An, and Q. Wen. A survey on trustworthy llm agents: Threats and countermeasures, 2025. 2

[25] Y. Zhao, J. Wang, L. Xiang, X. Zhang, Z. Guo, C. Turkay, Y. Zhang, and S. Chen. LightVA: Lightweight Visual Analytics With LLM Agent-Based Task Planning and Execution . *IEEE Transactions on Visualization & Computer Graphics*, 31(09):6162–6177, Sept. 2025. doi: 10.1109/TVCG.2024.3496112 2

[26] Y. Zhao, Y. Zhang, Y. Zhang, X. Zhao, J. Wang, Z. Shao, C. Turkay, and S. Chen. Leva: Using large language models to enhance visual analytics. *IEEE Transactions on Visualization and Computer Graphics*, 31(3):1830–1847, Mar. 2025. doi: 10.1109/TVCG.2024.3368060 2