AI時代的研究計畫撰寫：提升效率及說服力之實戰應用

中央研究院資訊科學研究所
黃瀚萱副研究員
2025
https://tinyurl.com/sinicalkt202511

Hen-Hsen Huang (黃瀚萱)

- Experiences
  - Associate Research Fellow, Institute of Information Science, Academia Sinica
  - Assistant Professor, Department of Computer Science, National Chengchi University
- Research areas
  - Natural language processing
  - Computational linguistics
  - Information retrieval
- Recent Projects:
  - Core model training for the TAIDE project
  - NTUH ICD-coder

Agenda

- Is it ethical to use LLMs in proposal/paper writing?
- The choice of LLMs
- General editing
- Related work survey
- Result analysis
- Proofreading

Publishers’ Policies

- Nature
  - https://www.nature.com/articles/s42256-023-00678-6
- Association for Computational Linguistics (ACL)
  - https://2023.aclweb.org/blog/ACL-2023-policy/

AI Policy of Nature

- AI Authorship
  - The use of an LLM (or other AI-tool) for “AI assisted copy editing” purposes does not need to be declared
  - Use of an LLM should be properly documented in the Methods section
  - Human accountability for the final version of the text and agreement from the authors that the edits reflect their original work.
- Generative AI images
  - While legal issues relating to AI-generated images and videos remain broadly unresolved, Springer Nature journals are unable to permit its use for publication.
  - With some exceptions

https://www.nature.com/nature-portfolio/editorial-policies/ai

Assistance Purely with the Language of the Paper

- When generative models are used for paraphrasing or polishing the author's original content, rather than for suggesting new content - they are similar to tools like Grammarly, spell checkers, dictionary and synonym tools, which have all been perfectly acceptable for years.
- If the authors are not sufficiently fluent to notice when the generated output does not match their intended ideas, using such tools without further checking could yield worse results than simpler-but-more-accurate English.
- The use of tools that only assist with language, like Grammarly or spell checkers, does not need to be disclosed.

Short-form Input Assistance

- Even though predictive keyboards or tools like smart compose in google docs are also powered by generative language models, nobody objected to them, since hardly anyone would try to use them to generate a long, unique and coherent text: it would simply not be practical.
- Similarly to language tools above, the use of such tools does not need to be disclosed in response to the writing assistance question.

Table Editing

Transpose the following latex table:

```
\begin{table}[tbh!]
    \centering
    \small
    \begin{tabular}{lccccc}
        \toprule
        Model & Accuracy & Precision & Recall & F-score
\\ %& Main\\
        \midrule
         Vicuna & 0.0426 & 0.0473 & 0.3012 & 0.0817 \\
         JointLAAT & 0.6150 & 0.7531 & 0.7702 & 0.7616
\\ %& 0.6284 \\
         Ours & 0.6738 & 0.8208 & 0.7885 & 0.8043 \\ % &
0.7686 \\
         % Ours & & & & & \\
         \bottomrule
    \end{tabular}
    \caption{Experimental results of full-code
prediction, reported scores are
micro-averaged.}\label{tab:overall}
\end{table}
```

Model	Accuracy	Precision	Recall	F-score
Vicuna	0.0426	0.0473	0.3012	0.0817
JointLAAT	0.6150	0.7531	0.7702	0.7616
Ours	0.6738	0.8208	0.7885	0.8043

Table 2: Experimental results of full-code prediction, reported scores are micro-averaged.

Metric	Vicuna	JointLAAT	Ours
Accuracy	0.0426	0.6150	0.6738
Precision	0.0473	0.7531	0.8208
Recall	0.3012	0.7702	0.7885
F-score	0.0817	0.7616	0.8043

Table 3: Experimental results of full-code prediction (transposed). Reported scores are micro-averaged.

Figure Generation

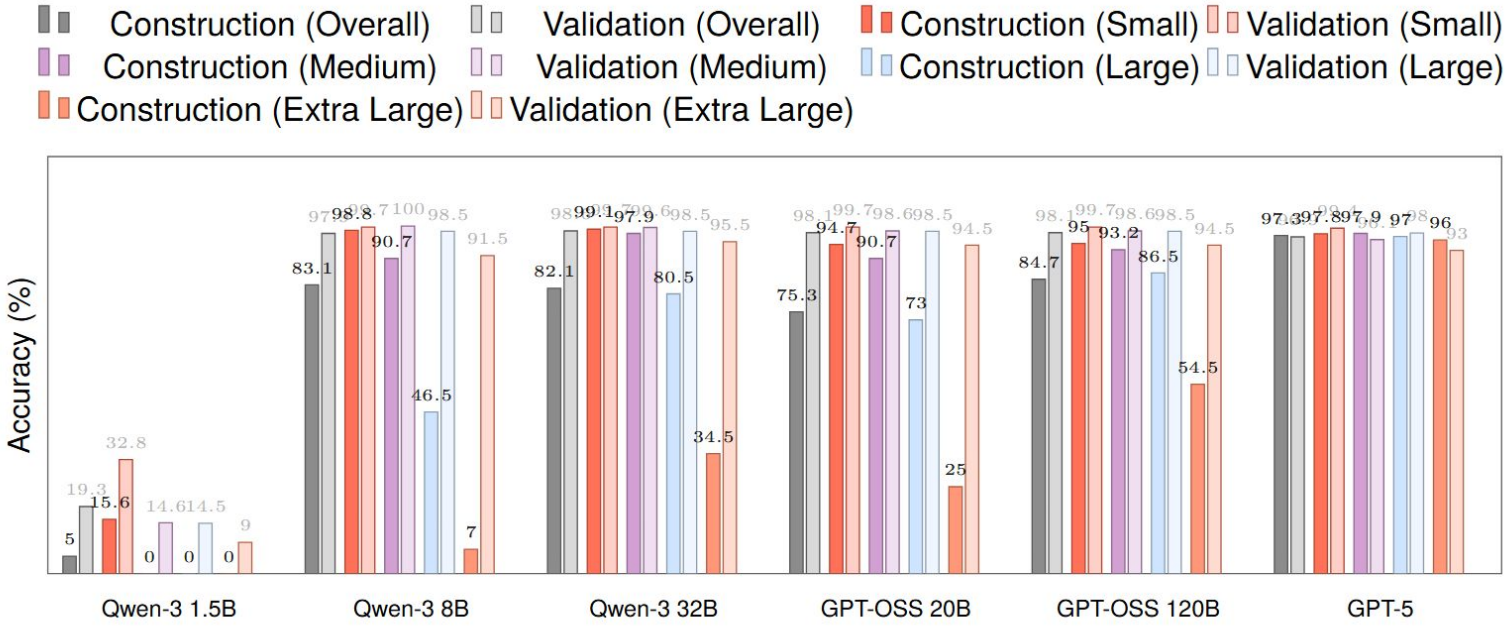


Figure 4: Accuracy of construction and validation on the ZebraLogic task across different solution space sizes (small, medium, large, extra large). While construction remains consistently high, validation accuracy decreases sharply with increasing logical complexity, especially for smaller models.

```
\begin{figure*}[t!]
\centering
\begin{tikzpicture}
\begin{axis}[
  legend columns=4,
  ybar,
  width=17cm,
  height=6cm,
  bar width=4pt,
  ymin=0, ymax=120,
  enlargelimits=0.1,
  enlarge y limits=0.0,
  ylabel={Accuracy (\%)},
  ylabel style={font=\footnotesize},
  % yticklabel=\pgfmathprintnumber{\tick}\%,
  yticklabel style={font=\scriptsize}, % show % on ticks
  ymajorticks=false,
  grid style={dashed,gray!35},
  symbolic x coords={Qwen-3 1.5B,Qwen-3 8B,Qwen-3 32B,GPT-OSS
20B,GPT-OSS 120B,GPT-5},
  xtick=data,
  xtick style={draw=none},
  xticklabel style={font=\scriptsize},
  nodes near coords,
  nodes near coords align={vertical},
  every node near coord/.append style={
    color=black,
    font=\tiny,
    /pgf/number format/fixed,

\addplot+[bar shift=-30pt, draw=gray!70!black, fill=gray, fill
opacity=0.9]
coordinates {
  (Qwen-3 1.5B,5.0)
  (Qwen-3 8B,83.1)
  (Qwen-3 32B,82.1)
  (GPT-OSS 20B,75.3)
  (GPT-OSS 120B,84.7)
  (GPT-5,97.3)
};

% Overall Judge
\addplot+[bar shift=-25pt, draw=gray!70!black, fill=gray, fill
opacity=0.3]
coordinates {
  (Qwen-3 1.5B,19.3)
  (Qwen-3 8B,97.9)
  (Qwen-3 32B,98.6)
  (GPT-OSS 20B,98.1)
  (GPT-OSS 120B,98.1)
  (GPT-5,96.9)
};
% Small Construction
\addplot+[bar shift=-18pt, draw=Tomato1!70!black, fill=Tomato1, fill
opacity=0.9]
coordinates {
  (Qwen-3 1.5B,15.6) (Qwen-3 8B,98.8) (Qwen-3 32B,99.1)
  (GPT-OSS 20B,94.7) (GPT-OSS 120B,95.0) (GPT-5,97.8)
};
```

Content Filtering

- Removing the header/footer of the response from LLMs

Here's a refined version of your contribution statement, keeping it precise, formal, and polished:

\item Our approach is theoretically equivalent to conventional beam search while substantially reducing memory and computational overhead. Empirical results across three modern transformer architectures—Multi-Head, Grouped Query, and Sliding Window Attention—demonstrate that it preserves output quality, with differences from conventional beam search being statistically insignificant.

Would you like me to make it **more concise** (for a bullet-style contribution list) or **more detailed** (closer to a full-sentence claim in a paragraph)?

- Removing all the inappropriate language usage
  - 旨在、信息、總的來說…

Literature Search

- Generative text models may be used as search assistants, e.g. to identify relevant literature.
- However, we expect the authors to read and discuss such references, just like the references identified by a regular search engine or a semantic literature recommendation tool.
- The usual requirements for citation accuracy and thoroughness of literature reviews apply; beware of the possible biases in suggested citations.

Literature Review

- Google Scholar
  - Recent, accurate works
- Allen AI Paper Finder
  - https://allenai.org/blog/paper-finder
- ChatGPT Deep Research
- Gemini
- Perplexity

User

Generate a survey on LLMs for text clustering

Researched for 39 seconds

View steps

Asta

While I didn't find a perfect match, I did find 7 relevant ones and 30 others.

This is what I searched for:

- Papers discussing: *Survey of large language models (LLMs) for text clustering.*
- Judged by the following relevance criteria:
  - *LLMs for Text Clustering*: The paper discusses the use of Large Language Models (LLMs) specifically for the task of text clustering. It should cover how LLMs are adapted, fine-tuned, or used in novel ways to perform text clustering.
  - *Survey/Review Focus*: The paper is a survey or review article that provides a comprehensive overview of existing research, methods, and techniques related to using LLMs for text clustering. It should synthesize and analyze multiple approaches rather than presenting a single, novel method.

You can either refine your query or start a new one.

You can ask me to "**work harder**" to run a more exhaustive search.

View 37 Papers

37 Papers

Large Language Models Enable Few-Shot Clustering

Vijay Viswanathan Kiril Gashteovski Carolin (Haas) Lawrence +1 Authors Graham Neubig

• Transactions of the Association for Computational Linguistics • 2023

Relevant: This paper explores using LLMs to improve text clustering, specifically focusing on semi-supervised methods. It tests LLMs in different stages of the clustering process: pre-clustering for feature enhancement, during clustering for constraint provision, and post-clustering for correction. While it builds on existing work, it introduces a novel approach rather than providing a broad survey.

LLMs for Text Clustering | Survey/Review Focus

Show Evidence

Cited by 90 Copy BibTeX

Large Language Models Offer an Alternative to the Traditional Approach of Topic Modelling

Yida Mu Chun Dong Kalina Bontcheva Xingyi Song

• International Conference on Language Resources and Evaluation • 2024

Relevant: The paper uses LLMs for text clustering by prompting them to generate topics from documents and evaluating their clustering efficacy. While it summarizes advantages and constraints, it primarily introduces a framework and experiments, not a broad survey.

LLMs for Text Clustering | Survey/Review Focus

Show Evidence

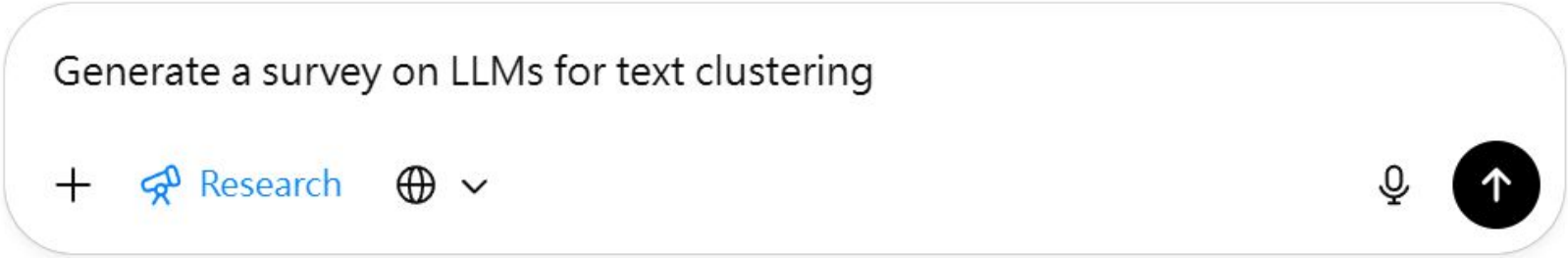
Cited by 36 Copy BibTeX

Context-Aware Clustering using Large Language Models

Sindhu Tipirneni Ravinarayana Adkathimar Nurendra Choudhary +4 Authors Chandan K. Reddy

• arXiv.org • 2024

Relevant: The paper focuses on using LLMs for text clustering, specifically addressing the underexplored potential of LLMs in this area. It also references recent works that use LLMs as oracles in the clustering process, making it somewhat relevant as a review.



To tailor the survey effectively, could you clarify a few things:

1. Who is the target audience? (e.g., NLP researchers, data scientists, industry professionals, students)
2. What is the primary goal of the survey? (e.g., understanding usage trends, evaluating performance perceptions, identifying common challenges)
3. How long should the survey be? (e.g., short with ~10 questions, or more detailed with ~20-30 questions)
4. Are you interested in specific types of LLMs or clustering applications (e.g., topic modeling, customer segmentation, document deduplication)?
5. Do you need different question types? (e.g., multiple choice, Likert scale, open-ended questions)

Once I have these details, I can generate a focused and effective survey for you.

ChatGPT Deep Research

ChatGPT

Large Language Models for Text Clustering: Recent Advances and Approaches

LLM Embeddings vs. Traditional Text Representations

Early text clustering methods relied on bag-of-words representations like TF-IDF, which treat each term independently and ignore context [1]. This often led to clusters that fail to capture semantic relationships between texts. The introduction of neural embeddings (e.g. Word2Vec, GloVe) and later contextual models like BERT brought significant improvements by encoding semantic and syntactic nuances in dense vectors [2] [3]. BERT-based sentence embeddings became a popular choice for clustering and topic modeling around 2020–2022, yielding more coherent groupings than TF-IDF or static embeddings [3]. For example, the BERTopic framework (Grootendorst 2022) uses BERT embeddings with clustering (UMAP + HDBSCAN) to discover interpretable topics, outperforming traditional LDA on semantic coherence [4].

Recent studies extend these approaches with **large language model (LLM) embeddings**. LLMs such as GPT, T5, and LLaMA generate high-dimensional embeddings that encode unprecedented breadth of linguistic knowledge [5]. Petukhova et al. (2024) benchmarked embeddings ranging from TF-IDF to OpenAI's GPT-3.5 (Ada-002) and LLaMA-2, evaluating their impact on clustering [6]. They found that **LLM-based embeddings excel at capturing nuanced language structure**, leading to purer and more meaningful clusters [7]. Notably, even without fine-tuning, embeddings from GPT-grade models improved clustering performance over "lightweight" models, although a strong baseline like BERT remained competitive among smaller models [8]. Interestingly, simply increasing embedding dimensionality or applying LLM-based text *summarization* as a preprocessing step did not uniformly improve clustering quality [8]. This suggests that more features or data compression aren't guaranteed benefits – the quality of representation matters more. There is a trade-off between rich, nuanced embeddings and computational feasibility: larger LLM embeddings yield finer semantic distinctions, but come at higher cost in memory and computation. Overall, the field has shifted from sparse vectors toward **LLM-augmented representations**, which consistently outperform traditional bag-of-words in capturing cluster structure [6].

Prompt-Based and Zero-Shot Clustering with LLMs

Beyond using LLMs as feature extractors, researchers are exploring **prompt-based, zero-shot clustering**, where the LLM's generative capabilities directly drive the grouping of texts. Huang and He (2024) propose reframing clustering as a *classification* task using LLMs [9]. In their framework, the model is first prompted to **generate candidate cluster labels** (topics) for the dataset, then asked to assign the best label to each sample, all in-context without training [10] [11]. This zero-shot method achieved clustering performance on par with state-of-the-art embedding+algorithm pipelines [11], while bypassing the need for specialized similarity metrics or iterative algorithms. It essentially leverages the LLM's world knowledge to hypothesize categories and categorize texts accordingly. Similarly, other work has used GPT-3/4 in zero-shot mode to cluster or categorize texts by giving instructions like "group these documents by topic" and letting the model infer clusters. These **LLM-as-clusterer** approaches are attractive for their simplicity, but they raise new questions. For instance, how many clusters should the model generate? How consistent are the results across multiple runs? And can the model resist trivial cues like writing style?

A case study on **multilingual news clustering** highlights some challenges. Schneider et al. (2024) used ChatGPT in a zero-shot setting to cluster news articles across languages [12]. They found the model largely grouped articles by language rather than by topic content, neglecting cross-lingual similarities [13]. This indicates that without careful prompt design or constraints, LLMs may latch onto surface features (e.g. language or style) when clustering, undermining the intended criteria. It underscores an ongoing debate: **LLMs have impressive generalization abilities, but guiding them to cluster on the "right" semantic basis remains non-trivial**. Some recent techniques attempt to mitigate this by providing exemplars or more detailed instructions (e.g. specifying that clusters should ignore language and focus on themes), which moves into the territory of few-shot prompting.

Few-Shot and Semi-Supervised Clustering with LLM Guidance

Another line of research integrates LLMs into the clustering process as intelligent oracles or data augmenters in a **semi-supervised** fashion. Instead of fully unsupervised clustering, these methods assume access to a small amount of expert knowledge (or allow minimal queries to an LLM) to greatly improve cluster quality. A representative example is **CLUSTERLLM** (Zhang et al., EMNLP 2023), which treats a large language model as a guide for clustering [14] [15]. Since API-based LLMs like ChatGPT do not expose their internal embeddings (and thus cannot be directly plugged into standard clustering algorithms) [16], CLUSTERLLM finds another way to exploit them. It uses an instruction-tuned LLM (ChatGPT) to answer *pairwise and triplet comparison questions* about the data, which provides feedback to adjust a smaller embedder's representations and to determine the appropriate cluster granularity [15] [17]. For example, given three samples A, B, C, the LLM is asked "does A belong with B rather than C?", to learn fine distinctions [18]. It also asks the LLM whether pairs of points should be in the same cluster to decide if clusters need to be merged or split [19]. These LLM-informed signals are then used to fine-tune the embeddings or set constraints in clustering. **Extensive experiments on 14 datasets** showed that CLUSTERLLM achieved consistently better clustering accuracy and NMI than purely unsupervised baselines, with only a handful of GPT queries (costing on the order of \$0.6 per dataset) [20]. This hybrid approach enjoys some benefits of LLM "wisdom" (leveraging its emergent semantic knowledge) without the cost of embedding every data point through a giant model.

In a similar vein, **LLM-assisted few-shot clustering** by Joshi et al. (2024) explores using LLMs at three points: *before* clustering (to improve features), *during* clustering (to provide pairwise constraints), and *after* clustering (to correct mistakes) [21] [22]. They found that incorporating an LLM in the first two stages yields significant gains in cluster quality [23]. Concretely, one method enriches each document's representation by prompting an LLM to generate descriptive *keyphrases*, which are then encoded and appended to the document embedding [24] [25]. This steers the clustering algorithm toward semantically important features (akin to an expert highlighting what aspects define clusters). Another method treats the LLM as a **pairwise similarity oracle**, where the user provides a few examples of similar vs. dissimilar pairs and the LLM generates additional "pseudo-label" constraints for the clustering algorithm [26] [27]. Using these LLM-generated constraints in a classical constrained k-means algorithm led to substantial improvement in clustering purity and adjusted Rand index on tasks like intent clustering and topic grouping [28]. Notably, the LLM-driven approach was able to approach the clustering performance of a human oracle (who fully labels pairwise relations) at a **fraction of the annotation cost** [29]. This demonstrates the potential of LLMs to *amplify* a small amount of expert guidance – effectively acting as a force multiplier in interactive clustering. A surprise finding was that even providing an **instruction** to the LLM about the clustering task (without any hand-picked examples) added significant value to the clustering outcome [30]. This insight opens up new possibilities of guiding clustering with high-level natural language instructions (e.g. "group news articles by underlying event, not by publication date") instead of traditional parameter tuning.

These semi-supervised techniques, however, come with practical considerations. They assume access to a strong LLM (often via API) during clustering, which introduces latency and cost for large datasets. There is also a risk that LLM-generated constraints could be inconsistent or reflect biases. Nonetheless, the research so far indicates that with careful prompt engineering and a few well-chosen examples, **LLMs can dramatically reduce the manual effort needed to achieve high-quality clusters**, bridging the gap between unsupervised discovery and supervised categorization [28] [29].

Evaluation Challenges: Coherence, Labeling, and Metrics

Evaluating text clustering is notoriously difficult, and the advent of LLM-based methods both helps and complicates matters. When ground-truth labels or categories are available for a dataset, researchers typically use **external metrics** like clustering accuracy, F₁-score, Adjusted Rand Index (ARI), or Normalized Mutual Information (NMI) to quantify how well clusters recover the known classes [31] [32]. For instance, many recent works report ARI and NMI to compare LLM-driven clustering with baselines [31]. These metrics, however, only apply in scenarios where a gold standard exists (e.g. intent categories, topic labels), which is often not the case in open-ended clustering problems.

In truly unsupervised settings, **intrinsic metrics** are used to gauge cluster quality without reference to labels. Common measures include Silhouette Score and Calinski-Harabasz Index, which assess how tight and well-separated the clusters are in the embedding space [33]. LLM-based representations often yield higher intrinsic scores than TF-IDF because they place similar meanings closer together, improving cohesion. Nonetheless, there is an ongoing debate about the reliability of intrinsic metrics – a high silhouette score doesn't always equate to semantically meaningful clusters. For example, an LLM might cluster documents by writing style or length (yielding well-separated groups) even if those clusters are not useful for a human analyst's needs.

Cluster coherence is a particularly important concept for evaluating topic-oriented clustering. It refers to how interpretable or semantically consistent the texts in a cluster are. Traditional topic models use coherence measures (like PMI or NPMI on top words) as proxies. In the LLM era, researchers are also using LLMs themselves to aid evaluation. One innovative approach is to have an LLM *name or summarize each cluster* and then judge the quality of those names. Preiss et al. (2024) studied using GPT-3.5 to automatically generate names for clusters of research abstracts and clinical notes [34]. Impressively, with a well-crafted prompt that provided key cluster information, the LLM's proposed labels were often as good as or better than human-generated labels on various quality criteria [35]. The best prompting strategy (e.g. giving the model a list of representative words from the cluster) produced cluster names that beat manual naming across all evaluated aspects [36]. This suggests LLMs can not only cluster texts but also help interpret clusters – a long-standing challenge in unsupervised learning. However, the same study noted that **results varied greatly with prompt design and dataset** [37]. In practice, one should try multiple prompt strategies and perhaps a human-in-the-loop check to ensure the generated labels make sense. The use of LLMs for cluster labeling is still emerging, but it hints at a future where cluster evaluation might include asking an LLM how coherent a cluster's theme is or how well a label describes it.

Google Gemini

2. Related Work

Our work is situated at the intersection of unsupervised text clustering and the emergent reasoning capabilities of large language models (LLMs). We position our contribution relative to three dominant paradigms in this rapidly evolving area.

2.1 LLMs as Feature Extractors

The most established approach for leveraging modern neural models in text clustering has been to use them as powerful feature extractors. This paradigm began with static embeddings like Word2Vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014), but became dominant with the advent of contextual encoders like BERT (Devlin et al., 2019). This line of work treats the text representation and the clustering algorithm as two distinct, sequential steps.

Recent advancements have largely focused on producing higher-quality embeddings. Frameworks like BERTopic (Grootendorst, 2022) have gained popularity by combining state-of-the-art sentence embeddings (Reimers & Gurevych, 2019) with density-based clustering algorithms like HDBSCAN (Campello et al., 2013) and a class-based TF-IDF for interpretability. The Massive Text Embedding Benchmark (MTEB) (Muennighoff et al., 2023) includes a dedicated clustering task, driving competition and rapid improvement in embedding models specifically for this purpose. While efficient and scalable, these methods' success is fully dependent on the pre-trained embedding's alignment with the user's desired clustering criteria, and cluster interpretability remains a separate, post-hoc challenge.

2.2 LLMs as Generative Clusterers

A more recent and disruptive paradigm reframes text clustering as a generative, in-context learning task, effectively eliminating the need for traditional vector-based algorithms. Pioneering work in this area (Huang et al., 2024) proposes a "clustering-as-classification" framework. This is often a two-stage process: (1) The LLM is first prompted to generate a set of candidate cluster labels by processing small batches of the input data. (2) The LLM is then prompted again to merge semantically equivalent labels and, finally, to classify each document into one of the generated labels.

This approach leverages the LLM's emergent reasoning and summarization capabilities to produce clusters that are *natively* interpretable, as each cluster is defined by a descriptive semantic label. However, these methods face significant challenges in practice: they are often computationally expensive and slow due to the high volume of required API calls, and they are constrained by the LLM's context window, making it difficult to process large documents or maintain a coherent global label space (Wang et al., 2024a).

2.3 LLMs as Hybrid Guides and Refiners

A third body of work seeks a middle ground, integrating LLM reasoning into the clustering pipeline to guide, refine, or interpret the results of more traditional methods. This hybrid approach aims to combine the scalability of embedding-based methods with the semantic nuance of LLMs.

Low-Novelty Text

- Some authors may feel that describing widely known concepts is a waste of their time and can be automated.
- They should specify where such text was used, and convince the reviewers that the generation was checked to be accurate and is accompanied by relevant and appropriate citations (e.g., using block quotes for verbatim copying).
- If the generation copies text verbatim from existing work, the authors need to acknowledge all relevant citations: both the source of the text used and the source of the idea(s).

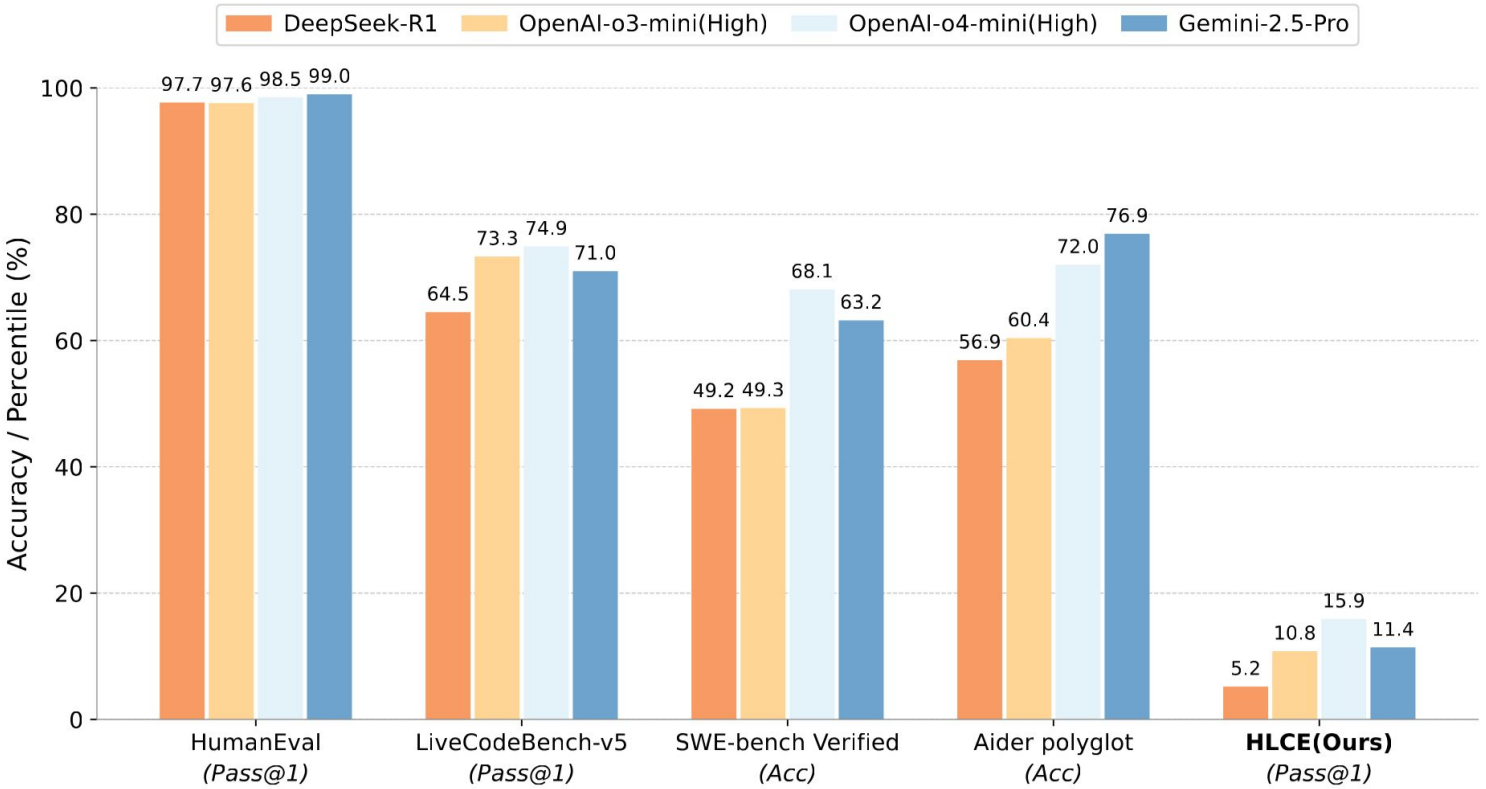
New Ideas

- If the model outputs read to the authors as new research ideas, that would deserve co-authorship or acknowledgement from a human colleague, and that the authors then developed themselves (e.g. topics to discuss, framing of the problem)
- We suggest acknowledging the use of the model, and checking for known sources for any such ideas to acknowledge them as well. Most likely, they came from other people’s work.

New Ideas + New Text

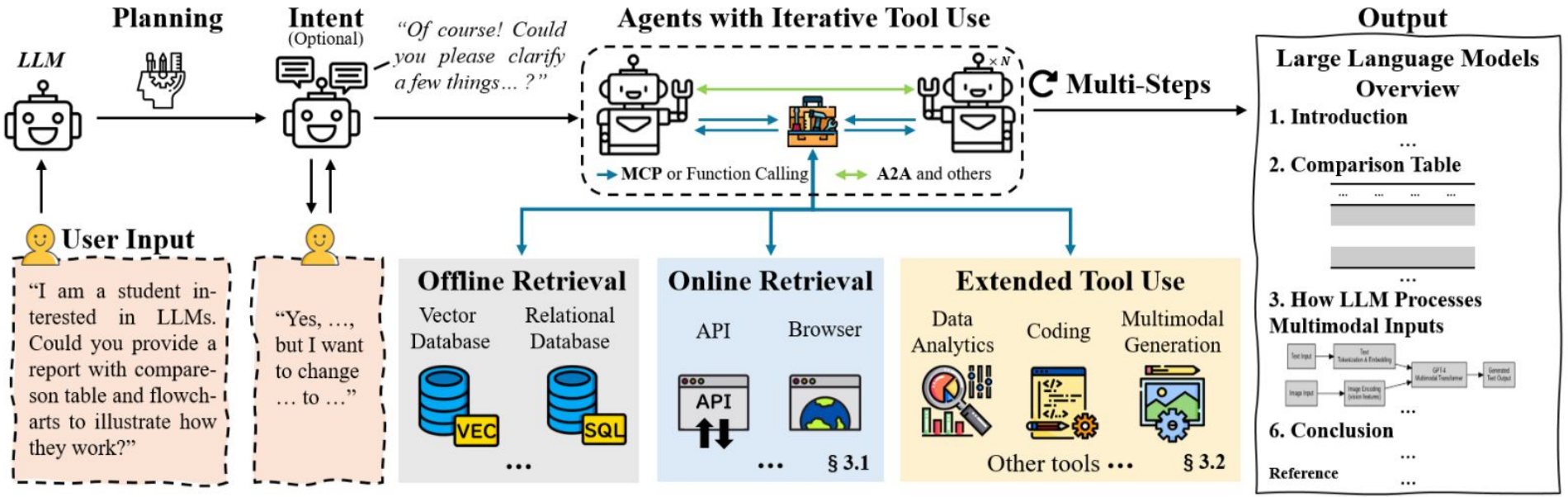
- A contributor of both ideas and their execution seems to us like the definition of a **co-author**, which the models cannot be. While the norms around the use of generative AI in research are being established, we would **discourage** such use in ACL submissions.
- If you choose to go down this road, you are welcome to make the case to the reviewers that this should be allowed, and that the new content is in fact correct, coherent, original and does not have missing citations.
- Note that, as our colleagues at ICML point out, currently it is not even clear who should take the credit for the generated text: the developers of the model, the authors of the training data, or the user who generated it.

Programming



https://arxiv.org/pdf/2506.12713v1

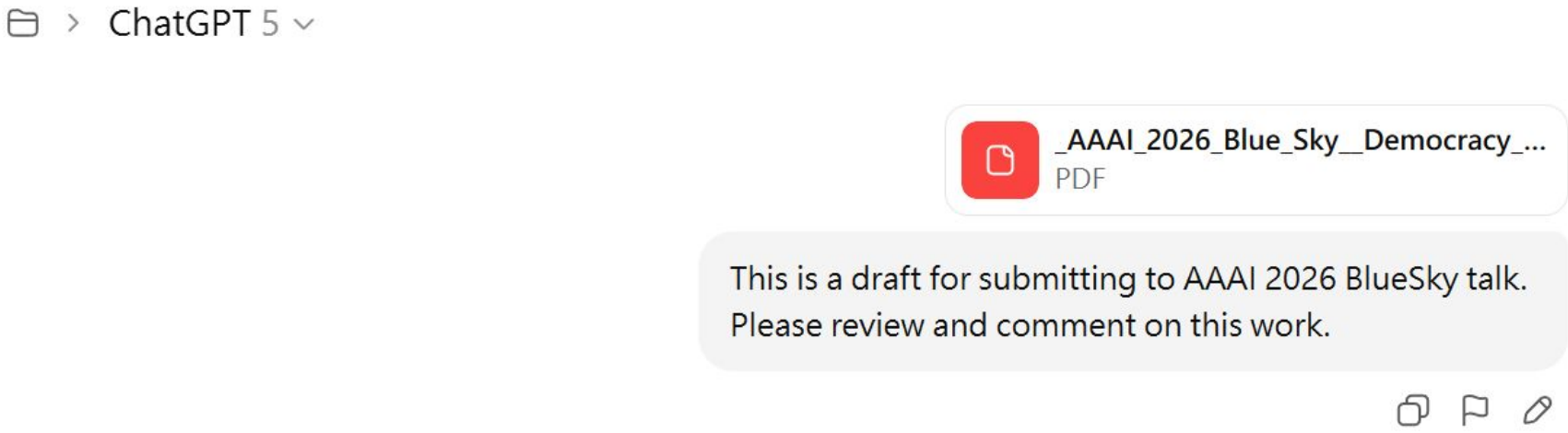
Doing Experiments



https://arxiv.org/pdf/2506.18096

Proofreading

- Check and review the draft that is about to submit



Strengths

1. **Clear positioning** – The paper sharply distinguishes between *hyperscale* and *small-to-medium* deployments. This structural divide is well-argued and gives the work a strong motivation _AAAI_2026_Blue_Sky__Democracy_... .
2. **Critical examination** – The analysis of MoE, speculative decoding, and complex RAG as "over-engineered" for non-hyperscale contexts is convincing and supported with references. The critique reads as both timely and constructive.
3. **Fresh agenda** – The call for simplicity, robustness, and deployability reframes efficiency research around *democratization* rather than *maximal performance*. This aligns well with Blue Sky's vision-setting goals.
4. **Breadth of methods** – The paper surveys a wide range of practical techniques: FlashAttention, GQA, LoRA, pruning, quantization, efficient decoding, KV cache compression, and lightweight RAG. This makes the proposal concrete, not just aspirational.
5. **Conceptual contribution** – Recasting *complexity itself as inefficiency* is a strong and quotable idea _AAAI_2026_Blue_Sky__Democra...

Weaknesses / Risks

1. **Audience targeting** – The draft says it is "directed at researchers in efficient LLMs" _AAAI_2026_Blue_Sky__Democracy_... , but much of the argumentation also speaks to practitioners/engineers in hospitals, finance, and government. Clarify:
   - Is this paper about guiding *research directions*?
   - Or is it also advocating *practices for organizations*?

   At times, it feels caught between the two. Blue Sky talks should prioritize **research agenda-setting**.
2. **Novelty vs. survey balance** – The paper is heavy on summarizing recent methods. While useful, it risks sounding like a *positioned survey* rather than a bold "blue sky" proposal. To strengthen originality:
   - Highlight *one or two unifying principles* (e.g., throughput-to-overhead ratio as a new efficiency metric) _AAAI_2026_Blue_Sky__Democracy_... .
   - Propose *research challenges* (e.g., designing efficiency methods with explicit constraints: "works on one GPU, deployable by a two-person IT team").
3. **Title framing** – Current title "Beyond Hyperscale: Rethinking Efficiency in LLMs for Small-to-Medium Deployments" is accurate, but perhaps not visionary enough for Blue Sky. A more ambitio... raming around *democratizing efficiency*

Suggestions

- **Clarify the reader**: Make it explicit this is a research agenda for the *efficiency research community*, not an engineering manual.
- **Lean into "democracy"**: The section "Research Agenda: Democracy in LLM Efficiency" is excellent. Consider pulling this framing earlier into the abstract and introduction to strengthen the hook.
- **Emphasize new metrics**: The *throughput-to-overhead ratio (TOR)* is a strong conceptual contribution. Make it central rather than buried late.
- **Visionary close**: The conclusion is solid but could be sharpened with a provocative challenge to the community, e.g., "The most impactful efficiency paper in 2030 will not reduce FLOPs by 20%, but will enable a rural hospital to deploy a safe LLM on one consumer GPU."

✅ Overall: This draft is strong and very close to Blue Sky expectations. It needs clearer **audience targeting** and more emphasis on **original conceptual contributions** (TOR metric, "complexity = inefficiency," democracy framing), while trimming down the survey-like detail.

Roles of LLMs in the Research Pipeline

- Conceptualization
- Investigation
- Implementation
- **Writing**
- Supervision
- Paper review
- Rebuttal