BordaRAG: Resolving Knowledge Conflict in Retrieval-Augmented Generation via Borda Voting Process

Yuxin Li
Gaoling School of Artificial Intelligence
Renmin University of China
Beijing, China
yuxin_li@ruc.edu.cn

Chen Xu
Gaoling School of Artificial Intelligence
Renmin University of China
Beijing, China
xc_chen@ruc.edu.cn

Jun Xu*
Gaoling School of Artificial Intelligence
Renmin University of China
Beijing, China
junxu@ruc.edu.cn

Ji-Rong Wen
Gaoling School of Artificial Intelligence
Renmin University of China
Beijing, China
jrwen@ruc.edu.cn

Abstract

Recently, research found that the documents retrieved from the Retrieval-Augmented Generation (RAG) may contain conflicting knowledge with each other, leading Large Language Models (LLMs) to generate incorrect responses. To solve such a problem, existing approaches usually only keep the most frequently mentioned knowledge from these documents, since they assume that the most representative knowledge aligns best with the true answer. Although effective in certain scenarios, these approaches often underperform when the most frequent knowledge is not the correct one. From the voting perspective, these methods can be regarded as a Majority Voting (MV) process, which chooses the most frequent candidates among different candidate knowledge. However, we show that the underperformance of such methods stems from that MV is only effective with a small number of candidates and binary voting scores. In contrast, in the RAG scenario, the candidates (knowledge) are very diverse, and the voting scores (document relevance scores) are typically continuous. Simply adapting MV in RAG will result in poor performance of LLMs. In voting theory, on the other hand, the preference-based voting methods represented by the Borda Voting (BV) consider the whole preference order of voters over all candidates, enabling the selection of candidates that better represent the collective viewpoint. Inspired by such an insight, we propose BordaRAG, a model designed to better select the most appropriate documents from conflicting documents. Specifically, BordaRAG first computes the preference scores of the documents over the candidate answers. After that, a BV component is designed to select the winning documents according to the preference scores. Finally, the chosen documents are provided to LLMs, which will generate the final response. Experimental results on three open-domain QA datasets show that BordaRAG can outperform all baselines.

CCS Concepts

• **Information systems** → **Information retrieval**.

Keywords

Retrieval-Augmented Generation, Knowledge Conflict, Social Choice, Borda Voting

ACM Reference Format:
Yuxin Li, Chen Xu, Jun Xu, and Ji-Rong Wen. 2025. BordaRAG: Resolving Knowledge Conflict in Retrieval-Augmented Generation via Borda Voting Process. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management (CIKM '25), November 10–14, 2025, Seoul, Republic of Korea.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3746252.3761038

1 Introduction

Retrieval-Augmented Generation (RAG) [5, 56] aims to retrieve documents from an external database to empower Large Language Models (LLMs) [1, 17]. Although RAG has been proven to be an effective way, recent studies have revealed that conflicting knowledge may exist among the retrieved external documents [8, 19, 51]. Supplying LLMs with such conflicting content can lead to inaccurate or unexpected answers [44, 51].

Recently, several approaches [29, 43, 49] have attempted to address this issue by filtering out conflicting documents and retaining only a single, presumably correct piece of knowledge. Typically, they often retain the most frequent knowledge among all documents, assuming that the dominant view likely reflects the correct answer. While intuitive and effective in reducing noise, this majority-based filtering strategy has a critical limitation: it equates popularity with correctness. In practice, documents holding the majority view may not necessarily be accurate, particularly in domains where misinformation is prevalent, minority opinions are underrepresented, or nuanced expertise is required. As a result, such methods risk discarding valuable minority evidence and reinforcing potentially biased or incorrect conclusions.

*Jun Xu is the corresponding author. Work partially done at Engineering Research Center of Next-Generation Intelligent Search and Recommendation, Ministry of Education.

CIKM '25, Seoul, Republic of Korea

ACM ISBN 979-8-4007-2040-6/2025/11
https://doi.org/10.1145/3746252.3761038

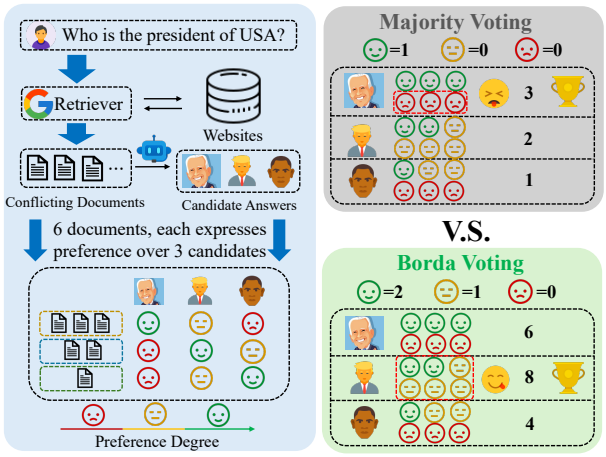


Figure 1: The comparison between the process of the Majority Voting and Borda Voting.

To better understand the limitations of existing methods, we reframe the task of selecting non-conflicting documents as a voting process. Specifically, each document can be viewed as a voter expressing preferences over possible answers (i.e., candidates). The goal is to aggregate these preferences to identify the most reliable answer. From such a voting perspective, previous methods can be interpreted as applying Majority Voting (MV) [3, 6, 37]—a classical mechanism in voting that selects the answer supported by the largest number of documents.

However, in voting research, MV often encounters the issue: when the number of candidates is large and the voters express their preferences in a non-extreme manner, it is often not very effective [10, 32]. Next, we will give an example to show such an insight. In Figure 1, the retriever retrieves six documents according to the query: *"Who is the president of U.S.A?"*. Among the six retrieved documents, there exist three candidate answers: *{Trump, Biden, and Obama}*. Among these answers, *Trump* is the correct answer, whereas the other two are incorrect. As shown in Figure 1, under the MV rule, only the most preferred candidates receive a positive score, while all others receive a score of zero. In such a way, the incorrect answer *Biden* wins during the voting since it is the most frequently preferred candidate answer. However, three documents strongly reject such a result, showing it is not a good result. On the other hand, Borda Voting (BV) takes into account the global preference distribution by assigning varying scores to candidates based on their ranked positions, thereby reflecting different levels of support. In such a way, the *Trump* wins by 8 points under the Borda voting mechanism, leading to a better voting result. Moving back to RAG scenarios, the candidate knowledge sources are highly diverse, and the voting scores (i.e., document relevance) are usually continuous rather than binary. As a result, previous methods may struggle to perform effectively in these settings.

In voting theory, the ineffectiveness of MV under a large number of candidates and continuous voting scores lies in its reliance *solely on each voter's top-ranked choice, ignoring the full spectrum of preferences*. This property can result in suboptimal voting outcomes (i.e., retaining incorrect documents), which ultimately degrades the performance of LLMs. In contrast, preference-based voting methods, which take into account the full preference distribution of each voter, provide a more robust and effective solution for handling knowledge conflicts in RAG scenarios. Our theoretical analysis in Section 4.4 also confirms that the preference-based voting methods exhibit superior theoretical performance compared to MV in RAG scenarios.

Motivated by the strengths of the preference-based voting method, Borda Voting [4, 6, 35], we introduce BordaRAG—a novel approach designed to resolve knowledge conflicts in the RAG process. By leveraging the global preference signals across all documents, BordaRAG enables LLMs to generate more accurate responses. The key components of BordaRAG include (1) **Preference Collection**: we utilize LLMs to generate answers for each retrieved document and then quantify the preference degree of each document toward all candidate answers; and (2) **Borda Voting**: then we aggregate document-level preferences through BV to filter out conflicting documents and ultimately derive the final answer. Based on the two components, BordaRAG can utilize the global preference of documents, leading to better performance under conflicting retrieved documents.

In summary, our key contributions are threefold:

(1) We formulate the RAG process as a voting process, and point out that the current MV-based methods will be ineffective under complex RAG scenarios.

(2) Based on the preference-based voting methods, we propose BordaRAG, a simple yet effective framework to select more representative documents under the knowledge-conflict situation.

(3) Our experimental results show our BordaRAG can outperform all the baselines under three widely used datasets.

2 Related Work

Retrieval-Augmented Generation. LLMs [1, 7, 17, 33, 40] perform well on tasks such as open-domain QA [30, 36, 55], but their limited parametric knowledge can cause inaccurate answers [11, 50]. RAG [5, 23, 24, 36] addresses this by incorporating external retrieved documents. A RAG system includes a retriever that selects relevant documents and a generator (LLM) that uses them to produce responses. However, previous research [8, 19, 21, 31, 51] finds that the retrieved documents may contain conflicting knowledge, leading to incorrect outputs.

Knowledge conflict RAG. Although RAG enhances LLMs with external documents, these may contain conflicting knowledge from diverse sources [19, 51], termed inter-context conflict [51]. To address this, some methods [46] apply query augmentation techniques and exploit the confidence of generated answers to mitigate conflicts among retrieved documents. Other methods [9, 43, 49] adopt a two-stage framework to handle conflicting knowledge in retrieved documents. In the first stage, an LLM extracts a perspective from each document independently. In the second stage, selection criteria identify the most representative perspective, with strategies differing across studies. For example, choosing the most frequent perspective [9, 49] or the one with the highest confidence score [43]. However, our analysis reveals that these methods implicitly rely

on an MV mechanism, which is only effective when the number of candidates is small and the preference scores are binary.

Voting mechanisms. Voting is essential for democratic governance and collective decision-making [6, 34, 37, 42, 48]. A voting process involves voters, candidates, and a mechanism, with MV being the most common. MV selects the candidate that appears most frequently as the top choice among voters. However, the Condorcet paradox [15, 28, 41, 52] shows that MV can be suboptimal. To address this, various methods have been proposed. According to the classification of Insua and French [20], there are two main types of methods to overcome the suboptimality of the MV mechanism. The first type, originally proposed by Condorcet [52], primarily focuses on pairwise comparisons between candidates and selects the one that wins the most head-to-head matchups as the winner. The second type, introduced by Borda [35], takes into account preference rankings of voters over all candidates and determines the winner based on an aggregated scoring system derived from these rankings.

In this paper, we adopt the second improved method (i.e., BV) to address the issue of suboptimality encountered in RAG systems.

3 Formulation

In this section, we introduce the notations used in RAG and voting processes, and present the correspondence between different symbols when formalizing RAG as a voting process.

3.1 Knowledge Conflict in RAG

A RAG system consists of two stages: retrieval and generation. In the retrieval stage, a retriever R is utilized to retrieve documents $\mathcal{D}_q$ from the document corpus according to the given query q: $\mathcal{D}_q = R(q)$. Among the documents $\mathcal{D}_q$, we abbreviate the number of retrieved documents $|\mathcal{D}_q|$ as n, which is a pre-defined number. Then the retrieved documents can be denoted as $\mathcal{D}_q = \{d_1, d_2, \cdots, d_n\}$, where d_i is the i-th document retrieved by the retriever. In the generation stage, we utilize a generator G, which is typically instantiated as a LLM, to generate an answer a based on the retrieved documents and the query: $a = G(\mathcal{D}_q, q)$.

However, the retrieved documents $\mathcal{D}_q$ may contain conflicting knowledge about the query q, which can lead the generator G to generating incorrect answers. We define the knowledge of each document d_i as: $a_i = G(q, d_i)$, which denotes the answer supported by the document d_i. When different documents support different answers, i.e., $a_i \neq a_j$ for $i \neq j$, the generator G will encounter a knowledge conflict problem when generating the final answer a.

To solve such a problem, previous methods may aggregate all retrieved documents to construct a new document set $\mathcal{D}'_q = F(\mathcal{D}_q)$, where F denotes the selecting function. F will select a new subset $\mathcal{D}'_q = \{d'_1, d'_2, \cdots, d'_k\}$ of original document set $\mathcal{D}_q$, where $k \leq n$ is the pre-defined number. Ideally, the subset $\mathcal{D}'_q$ will only contain the documents that have the knowledge related to the right answer. Then the generator G will generate the final answer a only based on the selected documents: $a = G(\mathcal{D}'_q, q)$.

3.2 RAG as Voting

A voting process mainly consists of three components: voters, candidates, and a voting mechanism. We define voters as a set $\mathcal{V}$ which contains n elements: $\mathcal{V} = \{v_1, v_2, \cdots, v_n\}$, where v_i denotes the i-th voter. We then define candidates as a set C which contains m elements: $C = \{c_1, c_2, \cdots, c_m\}$, where c_j denotes the j-th candidate.

Table 1: Mapping between fundamental RAG Components and the corresponding Voting Elements.

RAG Components	Voting Elements
Retrieved document d_i	Voter v_i
Knowledge of document a_i	Candidate c_i
Selection function $F(\mathcal{D}_q)$	Voting function $f(\mathbf{P})$

A voting process can be divided into two stages: preference representation and preference aggregation. In the stage of preference representation, we aim to construct a preference matrix $\mathbf{P} \in \mathbb{R}^{n \times m}$, where $\mathbf{P}_{i,j}$ denotes the preference degree of the i-th voter towards the j-th candidate. In the stage of preference aggregation, we aim to define a voting function $f(\mathbf{P}) = c^*$ to select a candidate winner c^* based on the pre-defined preference matrix $\mathbf{P}$.

For example, the voting function of MV can be denoted as:

$$f_{MV}(\mathbf{P}) = c_k,\ k = \underset{j \in \{1,2,\cdots,m\}}{\arg\max} \sum_{i=1}^{n} \mathbb{I}(j = \arg\max_k \mathbf{P}_{i,k}), \quad (1)$$

where $\mathbb{I}(\cdot)$ is the indicator function.

As shown in Eq. (1), the MV process counts the frequency of a candidate c_j being the top choice of voters $\mathcal{V}$, and then selects the candidate with the highest frequency to be the winner.

In this paper, we formulate the RAG process as a voting process. As shown in Table 1, a retrieved document d_i corresponds to a voter v_i, and its knowledge a_i corresponds to a candidate c_i. Since knowledge may conflict, voters aim to select documents that best answer the question q. The selection functions $F(\mathcal{D}_q)$ proposed by previous methods can be regarded as the voting function, denoted as $f(\mathbf{P})$. Therefore, the RAG process can be regarded as a voting process.

4 Method: BordaRAG

In this section, we aim to introduce our method for mitigating the problem of knowledge conflict in RAG systems. Our method mainly consists of two components: **Preference Collection** and **Borda Voting**. We first introduce the overall framework and then introduce the details of our proposed components.

4.1 Framework of BordaRAG

In order to mitigate the knowledge conflict in RAG systems, we propose a two-stage framework to reduce the degree of conflict in retrieved documents.

As shown in figure 2, the proposed BordaRAG consists of two components: **Preference Collection** and **Borda Voting**. In the stage of preference collection, we first prompt LLMs to generate answers for each retrieved document, and then deduplicate the generated answers to get a set of candidate answers. Next, we prompt LLMs to determine whether the document d_i supports the candidate answer a_j, and then utilize the probability of LLMs outputting the token "True" as the preference degree of voter v_i towards candidate c_j. The probability serves as $\mathbf{P}_{i,j}$ in the preference matrix $\mathbf{P}$. In the

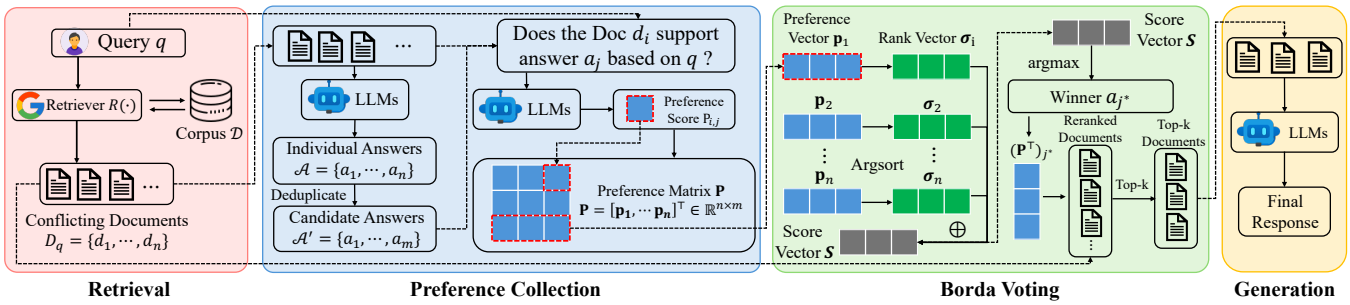


Figure 2: The overall framework of BordaRAG. The framework consists of two stages: preference collection, where retrieved documents are used to generate candidate answers and estimate preferences; and Borda Voting, where preferences are aggregated to generate the final answer.

Borda Voting stage, we first utilize the BV mechanism to select the winner according to the preference matrix. Then, we rerank the retrieved documents according to their preference degrees for the winner, and feed the top-k reranked documents to the LLM to generate the final answer.

Algorithm 1 illustrates the detailed steps of BordaRAG.

Algorithm 1 Algorithmic procedure of BordaRAG

Require: Query q, Retrieved Documents $\mathcal{D}_q$, Generator G
Ensure: Final Answer a

– Preference Collection –

Initialize Candidate Answer Set $\mathcal{A} \leftarrow \emptyset$
for $i = 1$ to n **do**
  $a_i \leftarrow G(d_i, q)$, $\mathcal{A} \leftarrow \mathcal{A} \cup \{a_i\}$
end for
$\mathcal{A} \leftarrow \text{Deduplicate}(\mathcal{A})$, $m \leftarrow |\mathcal{A}|$
Initialize Preference Matrix $\mathbf{P} \in \mathbb{R}^{n \times m}$
for $i = 1$ to n **do**
  for $j = 1$ to m **do**
    $\mathbf{P}_{i,j} \leftarrow \frac{\Pr[G(q,d_i,a_j)=\text{True}]}{\Pr[G(q,d_i,a_j)=\text{True}]+\Pr[G(q,d_i,a_j)=\text{False}]}$
  end for
end for

– Borda Voting –

Initialize Borda Score Vector $\mathbf{S} \in \mathbb{R}^m \leftarrow \mathbf{0}$
for $i = 1$ to n **do**
  $\tilde{\mathbf{p}}_i \leftarrow \text{softmax}(\mathbf{p}_i)$, $\sigma_i \leftarrow \text{argsort}(\tilde{\mathbf{p}}_i)$
  for $j = 1$ to m **do**
    $\mathbf{S}_j \leftarrow \mathbf{S}_j - \sigma_i(j) + m$
  end for
end for
$j^* \leftarrow \arg\max_{j \in [m]} \mathbf{S}_j$
$\mathcal{D}'_q = \{d_u \mid u \in \arg\max_{S \subset [n], |S|=k} \sum_{i \in S} \mathbf{P}_{i,j^*}\}$
$a \leftarrow G(\mathcal{D}'_q, q)$
return a

4.2 Preference Collection

Table 1 shows the mapping between RAG components and voting elements. However, the retrieved knowledge and preference matrix $\mathbf{P}$ are not explicitly represented. To address this, we propose a preference collection stage with two steps: (1) constructing candidate answers as voting candidates, and (2) constructing the preference matrix, which reflects the preference of voters.

Candidate answer construction. In the process of constructing the candidate answers set, we first prompt LLMs to generate answers for each retrieved document and remove duplicates. In this stage, semantically similar answers need not to be merged. In MV, such similar candidates (often referred to as *clone entities* in voting theory [12–14, 39]) may lead to vote splitting: since each voter can only cast one vote, some similar candidates may split the vote of voters who support this type of candidate. In contrast, preference-based voting mechanisms are less susceptible to this issue: similar candidates tend to appear close to each other in the rankings of the voters. As a result, clone entities tend to receive similar scores and may not cause the problem of vote splitting observed in MV.

Preference matrix construction. Constructing the preference matrix that captures document preferences over candidate answers is crucial for voting. Given a query q, a set of retrieved documents $\mathcal{D}_q = \{d_1, \cdots, d_n\}$, and a set of candidate answers $\mathcal{A} = \{a_1, \cdots, a_m\}$, we define the preference score of retrieved document d_i for candidate answer a_j as $s(d_i, a_j)$.

In order to estimate the preference score $s(d_i, a_j)$, we construct a set of query-document-answer triples (q, d_i, a_j). For each triple, we prompt LLMs to evaluate whether the document d_i supports the answer a_j under the context of the query q, and LLMs return a binary judgement: True (supports) or False (does not support). The prompt template can be found in Appendix B. The preference score of document d_i for answer a_j can be denoted as:

$$s(d_i, a_j) = \frac{\Pr[G(q, d_i, a_j) = \text{True}]}{\Pr[G(q, d_i, a_j) = \text{True}] + \Pr[G(q, d_i, a_j) = \text{False}]}. \quad (2)$$

The preference score $s(d_i, a_j)$ can qualify how likely the document d_i supports the candidate answer a_j according to the judgement of LLMs. Then, the preference of d_i over all candidate answers can be formulated as a preference vector $\mathbf{p}_i = [s(d_i, a_1), \cdots, s(d_i, a_m)] \in \mathbb{R}^m$. For n retrieved documents, we construct a preference matrix according to the preference vector of these documents. The preference matrix can be formulated as: $\mathbf{P} = [\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_n]^\top \in \mathbb{R}^{n \times m}$.

4.3 Borda Voting

In this section, we aim to utilize the BV mechanism to select the winner a_{j^*}. Based on the preference matrix $\mathbf{P} = [\mathbf{p}_1, \cdots, \mathbf{p}_n]^\top$ in Section 4.2, where $\mathbf{P}_{i,j}$ denotes the preference of document d_i over answer a_j, the BV method selects the candidate with the highest score as the winner:

$$a_{j^*} = f_{BV}(\mathbf{P}),\ j^* = \arg\max_{j \in [m]} \mathbf{S}_j,$$

where $[m] = \{1, 2, \cdots, m\}$, $\mathbf{S} \in \mathbb{R}^m$ is a score vector, and $\mathbf{S}_j$ denotes the overall score of candidate a_j. The score $\mathbf{S}_j$ is computed as:

$$\mathbf{S}_j = \sum_{i=1}^{n} m - \sigma_i(j),$$

where m is the total number of candidates, and $\sigma_i(j) : \{1, \cdots, m\} \to \{1, \cdots, m\}$ denotes the rank of candidate answer a_j in the preference order induced by document d_i.

After selecting the winning answer a_{j^*}, we sort the n retrieved documents $\mathcal{D}_q$ in descending order based on their preference scores $\mathbf{P}_{i,j^*}, i \in [n]$. Then we obtain the subset $\mathcal{D'}_q$ by selecting the documents that crosspond to the top-k largest elements of the preference scores $\mathbf{P}_{i,j^*}$, that is:

$$\mathcal{D}'_q = \{d_u \mid u \in \arg\max_{S \subset [n], |S|=k} \sum_{i \in S} \mathbf{P}_{i,j^*}\},$$

where k is a pre-defined number which denotes the number of selected documents. Then, we feed the documents in $\mathcal{D}'_q$ into generator G to generate the final answer:

$$a = G(\mathcal{D}'_q, q).$$

4.4 Theoretical Analysis

We provide a theoretical analysis of the upper bounds of expected distortion for MV and BV when selecting the most representative candidate answer in RAG.

Firstly, we use the expected distortion $\mathbb{E}_{\mathcal{A}\sim\theta}[D]$ to compare MV and BV. Formally, $\mathbb{E}_{\mathcal{A}\sim\theta}[D]$ is defined as:

$$\mathbb{E}_{\mathcal{A}\sim\theta}[D] = \mathbb{E}_{\mathcal{A}\sim\theta}\left[\frac{c(w(\mathcal{A}))}{c(o(\mathcal{A}))}\right],$$

where θ is the distribution of voters in the space by a density function and $c(a_j)$ denotes the degree of dissatisfaction of all voters towards candidate a_j. $w(\mathcal{A})$ deonotes the winner of candidates $\mathcal{A}$ selected by the voting mechanism $w(\cdot)$, $o(\mathcal{A})$ denotes the optimal candidate of $\mathcal{A}$: $o(\mathcal{A}) = \arg\min_{j \in [1,m]} c(a_j)$.

Expected distortion is usually used to measure the degree of deviation of a voting mechanism from efficiency. It reflects the gap between voting results and social optimal results, that is, the degree to which collective decision-making deviates from the optimal choice due to imperfect mechanisms or incomplete information.

Then, we derive the following theorem:

Theorem 1. *The expected distortion of MV is upper bounded by super-constant:*

$$\mathbb{E}_{\mathcal{A}\sim\theta}\left[\frac{c(w_{MV}(\mathcal{A}))}{c(o(\mathcal{A}))}\right] \in \omega(1),$$

while the expected distortion of BV is upper bounded by a constant:

$$\mathbb{E}_{\mathcal{A}\sim\theta}\left[\frac{c(w_{BV}(\mathcal{A}))}{c(o(\mathcal{A}))}\right] \in O(1).$$

The detailed proof of Theorem 1 can be found in Appendix A. Theorem 1 shows that with the increase of the number of candidate answers (i.e., conflicting knowledge in retrieved documents), the expected distortion of prior MV-based methods grows super-constantly. In contrast, the expected distortion of BordaRAG admits a constant upper bound.

Therefore, Theorem 1 shows that MV is only effective when the number of candidates is small. As the number of candidates increases, the expected distortion of MV, which relies on binary scores to select candidates, grows rapidly. In contrast, BV assigns continuous scores to candidates and provides a more comprehensive selection, resulting in a constant upper bound on the expected distortion. Therefore, BV emerges as a superior method for addressing the issue of knowledge conflict.

5 Experiments

In this section, we demonstrate the effectiveness of BordaRAG in addressing the problem of knowledge conflict. We begin by introducing the detailed experimental settings. Then, we compare the performance of BordaRAG with several baselines across multiple QA datasets. Finally, we provide an in-depth analysis of the experimental results. Our code can be found in https://github.com/RUC-YuxinLi/BordaRAG.

Table 2: Statistical information of three QA datasets.

Dataset	#Queries	Domain
NQ	91k	General knowledge (Google queries)
PopQA	14k	Popular entities (Wikipedia)
TriviaQA	174k	Web and Wikipedia

5.1 Experimental Settings

Datasets. We conduct our experiments on three QA task datasets: NQ, PopQA, and TriviaQA.

- **NQ** [25] is constructed from real anonymized Google Search queries. NQ is considered challenging due to its realistic question distribution and significant lexical variation between questions and answers.
- **PopQA** [27] targets factual questions about popular entities and is designed to evaluate long-tail and open-domain factual knowledge in LLMs.
- **TriviaQA** [22] contains complex questions that require reasoning over multiple pieces of evidence.

Table 2 summarizes the statistics of the three datasets. Following [43], we utilize the Google Search API [16] to retrieve the top 10 results for each query, extracting titles and snippets to build the retrieval corpus. Previous work [19, 43] has shown that conflicting knowledge often exists in retrieved documents, even within a single source. Our analysis in Section 5.3 confirms that documents retrieved from Google Search frequently contain multiple candidate

Table 3: Comparison of different methods on NQ, PopQA, and TriviaQA. Experimental results are reported in terms of accuracy on each dataset. Bold indicates the best performance, while underlined values denote the second-best. The results marked with $*$ indicate the improvements are statistically significant ($p < 0.05$). Our BordaRAG outperforms baselines across all datasets.

Type	Baseline	Llama3-8B-Instruct			Qwen2-7B-Instruct			Mistral-7B-Instruct		
		NQ	PopQA	TriviaQA	NQ	PopQA	TriviaQA	NQ	PopQA	TriviaQA
No-RAG	LLM [17]	0.269	0.250	0.659	0.262	0.221	0.597	0.310	0.274	0.697
	GenRead [53]	0.374	0.327	0.740	0.307	0.248	0.640	0.375	0.317	0.725
Prompt-based	InstructRAG [45]	<u>0.490</u>	0.395	0.838	0.480	0.400	<u>0.830</u>	<u>0.497</u>	<u>0.413</u>	0.824
	DAA [18]	0.437	0.370	0.739	<u>0.484</u>	0.404	0.822	0.485	0.398	0.823
MV-based	RobustRAG [49]	0.467	0.411	0.841	0.471	0.394	0.824	0.444	0.390	0.827
	AstuteRAG [43]	0.459	<u>0.439</u>	0.817	0.329	0.307	0.633	0.451	0.378	0.751
	USC [9]	0.477	0.409	<u>0.842</u>	0.483	<u>0.404</u>	0.825	0.495	0.410	<u>0.839</u>
Ours	**BordaRAG**	**0.535***	**0.458***	**0.865***	**0.502***	**0.417**	**0.852***	**0.522***	**0.420**	**0.850**
	Improvement	+9.2%	+4.3%	+2.7%	+3.7%	+3.2%	+2.6%	+5.0%	+1.7%	+1.3%

answers for a single query. Due to computational constraints, we sample 1000 queries from each dataset for experimentation.

Metrics. Following [43, 45, 49], we consider an answer to be correct if it contains the ground-truth answer. Formally, let $\mathcal{G}_q$ be the set of ground truth answers for q. The accuracy is:

$$\text{Acc} = \frac{1}{|Q|} \sum_{q \in Q} t(a, \mathcal{G}_q),$$

where $t(a, \mathcal{G}_q)$ checks whether there exists an $a_g \in \mathcal{G}_q$ such that the generated answer a contains the ground-truth answer a_g. It returns 1 if such an a_g exists, and 0 otherwise.

Settings of LLMs. We adapt LLaMA3-8B-Instruct [17], Qwen2-7B-Instruct [38] and Mistral-7B-Instruct-v0.3 [2] as generators in RAG. All models are served via the vLLM inference engine [26], with the decoding temperature set to 0 and top-p set to 0.8 to ensure deterministic generation with moderate sampling flexibility.

Baselines. We compare BordaRAG with methods aimed at improving robustness and reducing the impact of conflicting external knowledge to generate more reliable responses.

- **GenRead** [53] utilizes the internal knowledge of LLMs instead of external documents to enhance the RAG system.
- **USC** [9] samples multiple responses from the LLM and then aggregates them based on majority consensus. We set the sampling temperature to 0.7 during repeated sampling in USC.
- **InstructRAG** [45] prompts LLMs to generate responses according to the retrieved documents.
- **DAA** [18] prompts LLMs to evaluate the conflicting degree of the retrieved documents and ignore the noisy documents during the generation process.
- **RobustRAG** [49] aggregates the keywords of individually generated responses of each retrieved document, and prompts LLMs to answer the query using the filtered keywords, aiming to produce robust and trustworthy responses.
- **AstuteRAG** [43] clusters retrieved documents by underlying knowledge and selects the answer with the highest LLM-assigned confidence. As a CoT-based method, it may fail to identify the final answer as instructed, so we use Llama3-8B-Instruct to extract the final answer for fair comparison.

To ensure a fair comparison, we do not include variants of these methods that involve additional training or in-context learning.

5.2 Main Results

We evaluate our method on three datasets (NQ, PopQA, TriaviaQA) and three LLMs (Llama3-8B-Instruct, Qwen2-7B-Instruct, Mistral-7B-Instruct), using identical retrieval results for fairness. The experimental results are shown in Table 3, where bold indicates the best performance, while underlined values denote the second-best.

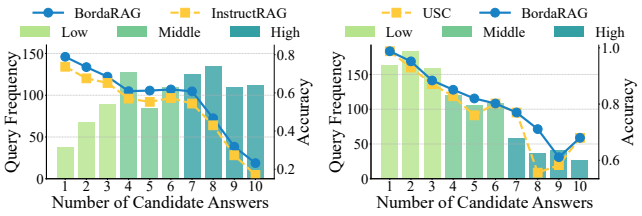
Firstly, we can observe that our method can outperform the baselines that utilize the internal knowledge of LLMs. More importantly, we find that our method can consistently outperform all the baselines that aim to select the appropriate documents/knowledge, including both MV-based methods and Prompt-based methods for LLMs under all datasets. This is because our method can select more appropriate documents/knowledge using the Borda voting process. While most other baselines only adapts the majority voting process, which only effective under a small number and binary voting scores, and the Prompt-based method DAA, which also cannot select the knowledge related to the right answer well.

Secondly, the improvements of our method are more substantial on NQ and PopQA, whereas the performance gains on TriviaQA are not that surprising. This is because in NQ and PopQA, most queries are associated with retrieved documents containing more than six candidate answers, reflecting a high degree of knowledge conflict. Our method shows notable improvements on such datasets. However, in TriviaQA, most queries correspond to documents with at most three candidate answers, indicating limited conflict, where the performance gains of BordaRAG are less pronounced.

5.3 Experimental Analysis

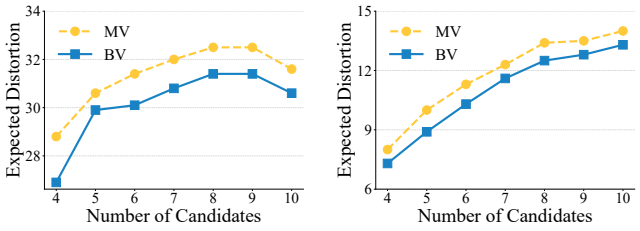
In this section, we analyze how the degree of conflict in retrieved documents relates to the characteristics of NQ and TriviaQA when using Llama3-8B-Instruct. Similar patterns are observed across other datasets and LLMs.

We compare BordaRAG with the best-performing baselines, InstructRAG and USC, across varying numbers of candidate answers



(a) NQ (b) TriviaQA

Figure 3: Query distribution and accuracy comparison of BordaRAG against top-performing baselines under varying numbers of candidate answers. In both panels, the x-axis indicates the number of candidate answers, the left y-axis shows the number of corresponding queries, and the right y-axis presents the average accuracy under each setting.



(a) NQ (b) TriviaQA

Figure 4: Comparison of expected distortion between MV and BV.

in the retrieved documents, and then present a case study highlighting its differences from MV-based methods in candidate selection.

Analysis on candidate numbers. Figure 3 illustrates the query distribution by candidate answer count in retrieved documents and compares our method with top baselines on NQ and TriviaQA. In both panels, the x-axis denotes the number of candidate answers. The left y-axis shows the number of queries with the corresponding candidate size, while the right y-axis reports the average accuracy of BordaRAG and the strongest baseline on those queries. We consider queries with no more than 3 candidate answers among the 10 retrieved results as having a low level of conflict, those with 4 to 6 candidate answers as medium conflict, and those with 7 or more as high conflict.

As shown in Figure 3 (a), most queries in NQ have more than six candidate answers, indicating a high conflicting level. This likely arises because NQ questions come from real Google queries, including both objective ones with low conflict and subjective ones with higher conflict. Figure 3 (a) also shows that accuracy drops as conflict increases, but our method consistently outperforms InstructRAG. This advantage stems from its ability to generate concise answers under low-conflict settings and to effectively filter documents in high-conflict scenarios.

Figure 3 (b) shows that the retrieval results of TriviaQA have lower conflict than that of NQ, since TriviaQA mainly involves fact-based questions that yield more consistent knowledge across documents. The accuracy results in Figure 3 (b) exhibit a similar trend to those in Figure 3 (a), showing a decline as conflict increases. At low conflict levels, BordaRAG performs comparably to USC, but its advantage over the baseline grows with higher conflict. These findings highlight the effectiveness and robustness of our method.

Comparison of expected distortion between MV and BV. In this section, to verify our theoretical analysis in Section 4.4, we aim to measure the expected distortion of utilizing BV and MV.

In Figure 4, we compare the expected distortion of MV and BV. Subfigure (a) shows the results on NQ, and subfigure (b) on TriviaQA. The x-axis indicates the number of candidates, and the y-axis the expected distortion, with the yellow and blue lines representing MV and BV, respectively.

In this experiment, we define distortion as the gap between each method's performance and the upper bound, i.e., the proportion of queries whose context contains the ground truth. The expected distortion is the mean distortion over queries with fewer candidate answers than the current value. MV results are obtained by replacing the BV function in our method with MV.

As shown in Figure 4, the expected distortion of MV is consistently higher than the distortion of BV among queries with different candidate numbers on two datasets, which means BV achieves better performance than MV under various conflicting levels. Therefore, the experimental results verify our theoretical analysis in Section 4.4.

Case study. To demonstrate BordaRAG in complex cases, Figure 5 shows a case study comparing BordaRAG with MV-based methods on the query *"rosie and the originals angel baby release date?"*, where ten documents are retrieved and candidate answer frequencies are derived using Llama3-8B-Instruct.

As shown in Figure 5, the answer *"1964"* appears most frequently (4 times) but it is a hallucination, while the correct answer *"1960"* (including *"November 1960"*) appears only twice. MV-based methods may select *"1964"* as the winner and prioritize documents supporting this incorrect answer, potentially reinforcing irrelevant or false knowledge. In contrast, BordaRAG selects *"1960"* based on global preference distribution and reranks documents by their support for it. As a result, BordaRAG identifies the correct answer and avoids the failure cases of MV-based methods.

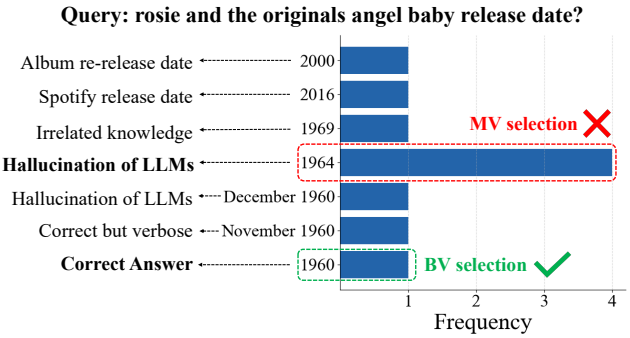


Figure 5: A case study comparing BV and MV in terms of their ability to identify the most representative knowledge.

5.4 Ablation Study

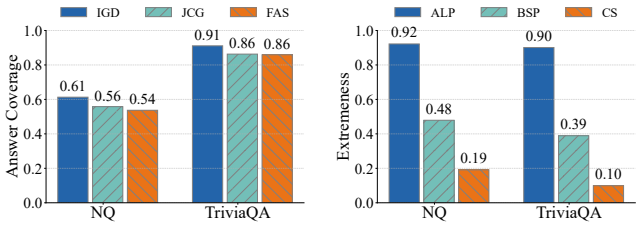
In this section, we analyze different BordaRAG components on NQ and TriviaQA with Llama3-8B-Instruct, and examine the efficiency–accuracy trade-off under varying k.

Methods of constructing candidate answers. To determine which method is more effective for constructing the candidate answer set, we experiment with three approaches using Llama3-8B-Instruct: (1) Independent Generation with Deduplication (**IGD**) [49]: generates an answer for each retrieved document and deduplicates the results; (2) Joint Contextual Generation (**JCG**) [43]: inputs all retrieved documents together and generates candidate answers in one pass; (3) Filtering Answer Set (**FAS**): applies LLMs to filter unlikely answers from the set produced by IGD.

Figure 6 (a) compares answer coverage of IGD (blue), JCG (green), and FAS (orange) on NQ (left) and TriviaQA (right). Coverage is defined as containing the ground-truth answer within the candidate set. As shown in Figure 6 (a), IGD achieves the best performance on NQ and TriviaQA. We attribute this to the fact that, when confronted with conflicting knowledge, LLMs tend to ignore less frequently mentioned knowledge. However, the ignored knowledge may be factually correct. Both JCG and FAS rely on LLMs to generate the candidate answer set, whereas IGD utilizes a rule-based approach to construct the answer set. This rule-based strategy prevents the less frequently mentioned but factually correct knowledge from being ignored by LLMs.

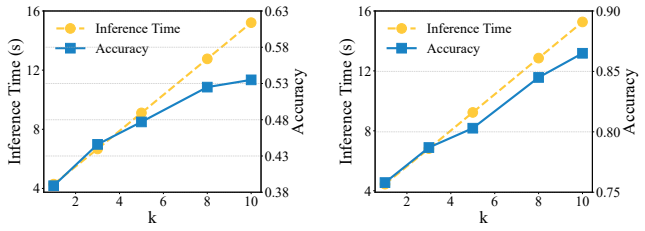
Methods of estimating preference scores. To determine which method can produce high-quality preference vectors, we experiment with three approaches: (1) Average Log-Probability (**ALP**) [47]: uses the average log-probability that a document assigns to a candidate answer as the preference score. (2) Confidence Scoring (**CS**) [43]: prompts the LLM to produce a confidence score for an answer given a document, which is used as the preference score. (3) Binary Support Probability (**BSP**) [54]: asks the LLM whether a document supports a candidate answer, using the probability of generating a "True" token as the preference score.

To evaluate the quality of generated preference vectors, we define a metric to measure the uniformity of a preference vector, named *extremeness*. In the context of voting, if all preferences are extremely skewed (e.g., only one candidate is strongly supported while all others receive low scores), the system is prone to bias. On the other hand, if all preferences are equal, it is hard to select a winner. A high-quality preference vector should reflect differences among candidates in a smooth and balanced manner.



(a) Candidate construction (b) Preference estimation

Figure 6: Evaluation of different methods for constructing candidate answers and estimating preference scores.



(a) NQ (b) TriviaQA

Figure 7: Trade-off between accuracy and efficiency.

Given a preference vector of the i-th voter: $\mathbf{p}_i = [p_1, p_2, \cdots, p_m]$, where p_j describes the preference of the voter of the j-th candidate answer. The preference vector $\mathbf{p}_i$ satisfies the normalization condition: $\sum_{j=1}^{m} p_j = 1$. we define the *extremeness* of $\mathbf{p}_i$ as:

$$\mathrm{Ext}(\mathbf{p}_i) = 1 - \frac{H(\mathbf{p}_i)}{\log m},$$

where $H(\mathbf{p}) = -\sum_{j=1}^{m} p_j \log p_j$ is the Shannon entropy.

The value of *extremeness* ranges from 0 to 1. *Extremeness* equaling to 0 indicates that the voter has equal preference for all candidates, and *extremeness* equaling to 1 indicates that the voter only supports one candidate while completely disregarding other candidates.

In Figure 6 (b), we present the comparison of the *extremeness* of three methods. The blue bars denote the results of ALP, the green bars denote the results of BSP, and the orange bars denote the results of CS. The left bars denote the results of the three methods on NQ, and the right bars denote the results on TriviaQA.

As shown in Figure 6 (b), the *extremeness* of ALP is close to 1 on both datasets, indicating that its preference vectors are overly sharp, meaning that each voter supports only one candidate and ignores the others. In contrast, CS exhibits an *extremeness* value near 0, indicating that its vectors are overly smooth, in the sense that each voter shows nearly equal preference for all candidates. From a voting perspective, both methods fail to capture meaningful distinctions in vector preferences. BSP, with *extremeness* around 0.5, offers a balanced representation, making it more suitable for estimating preference scores.

Trade-off between accuracy and efficiency. We study the efficiency–accuracy trade-off by varying the number of Top-k reranked documents in BordaRAG. As shown in Figure 7, the x-axis indicates the number of input documents k, the blue line shows average accuracy, and the yellow line shows average inference time, measured using the vLLM engine [26].

Figure 7 (a) shows the relationship between inference time and accuracy with varying k on NQ, while Figure 7 (b) shows the results on TriviaQA. Both figures exhibit a similar trend: as k increases, both accuracy and inference time rise. This highlights a trade-off: more retrieved documents improve answer quality but lead to longer inputs and lower efficiency.

6 Conslusion

In this paper, we formulate the methods that aims to solve the knowledge conflict in RAG as a voting process, where the retrieved documents are treated as voters and the knowledge they contain as candidates. Then previous methods aiming to address knowledge conflict can be viewed as utilizing the MV strategy, which is not effective under complex RAG scenarios. To address this, we propose BordaRAG, which utilizes the BV to select more appropriate documents to avoid the knowledge conflict. We provide both theoretical and empirical evidence showing that BordaRAG outperforms previous methods.

In the future, we plan to explore more effective voting strategies for resolving knowledge conflict in RAG, aiming to ensure that the selected winners are optimal.

Acknowledgments

This work was funded by the National Natural Science Foundation of China (No. 62472426), Beijing Key Laboratory of Research on Large Models and Intelligent Governance, fund for building world-class universities (disciplines) of Renmin University of China.

Appendix

A Proof of Theorem 1

In this section, we prove that when RAG is formalized as a voting system, the upper bound of the expected distortion for BV is less affected by the number of candidates compared to that for MV. First, we define a voting system with properties analogous to those of RAG. Then, inspired by [10], we deduce the properties of this voting system, which in turn lead to the proof of Theorem 1.

As shown in Table 1, retrieved documents $\mathcal{D}_q = \{d_1, \cdots, d_n\}$ serve as voters $\mathcal{V}$, and candidate answers $\mathcal{A} = \{a_1, a_2, \cdots, a_m\}$ as candidates C. We define the embedding model of LLMs as $e(\cdot)$, and the embedding of d_i and a_j can then be denoted as $e(d_i), e(a_j)$.

We define $e(d_i), e(a_j)$ in the same vector space Ω, a metric space (Ω, d) with $d_{\omega_1,\omega 2}$ denoting the distance between $\omega_1, \omega_2 \in \Omega$. The voter distribution in this space is given by a density function θ.

Based on the definition in Section 4.4, we have:

Definition 1. *A RAG system can be formalized as a non-strategic, representative, positional voting system.*

To better understand this definition, we provide further explanations of the three core properties:

- **Non-strategic**: As the LLM estimates preferences based on a single voter-candidate pair at a time, the process is inherently non-strategic.
- **Representative**: Since candidates are extracted from documents (voters), they can be viewed as representative of the voters.
- **Positional voting**: Both MV and BV are positional voting methods, where voters rank candidates by non-decreasing distance.

Inspired by [10], we construct a scoring function for such a non-strategic, representative, positional voting system:

Definition 2. *Let $\mathcal{V}$ be a voting system with m candidates and n voters, the total score σ of a_j is: $\sigma(a_j) = \int_\omega g_m(\pi_\omega(a_j))\theta_\omega \mathrm{d}\omega$, where $g_m : \{0, \cdots, m-1\} \to [0,1]$ is a non-increasing function with $g_m(0) = 1$ and $g_m(n-1) = 0$, $\pi_{d_i}(a_j)$ denotes the rank of a_j by d_i.*

For positional voting systems, the winning candidate is the one with the highest total score: $w(\mathcal{A}) \in \arg\max_{j\in[1,m]} \sigma(a_j)$.

Inspired by [10], we define the limit scoring rule:

Definition 3. *Let $\mathcal{V}$ be a positional voting system with m candidates, if exists a threshold m_0 satisfying:*

$$\forall x \in [0,1], \forall m > m_0, \forall \epsilon > 0, g_m(\lfloor x(m-1) \rfloor) \geq g(x) - \epsilon,$$

$$\forall x \in [0,1], \forall m > m_0, \forall \epsilon > 0, g_m(\lceil x(m-1) \rceil) \leq g(x) + \epsilon.$$

then $g : \mathbb{Q} \cap [0,1] \to [0,1]$ is the limit scoring rule of $\mathcal{V}$.

Based on Corollary 3.2 in [10] and our Definition 1,3, we derive the following lemma:

Lemma 2. *Let $\mathcal{V}$ be a non-strategic, representative positional voting system with limit scoring rule g, then:*

$$(1)\ \exists x_1, x_2 \in (0,1), g(x_1) \neq g(x_2) \Rightarrow \mathbb{E}_{\mathcal{A}\sim\theta}\left[\frac{c(w(\mathcal{A}))}{c(o(\mathcal{A}))}\right] \in O(1),$$

$$(2)\ \forall x \in (0,1), g(x) \equiv c, c \neq 1 \Rightarrow \mathbb{E}_{\mathcal{A}\sim\theta}\left[\frac{c(w(\mathcal{A}))}{c(o(\mathcal{A}))}\right] \in \omega(1).$$

where $\omega(1)$ indicates that the original expression has a super-constant upper bound, while $O(1)$ indicates that it has a constant upper bound.

Based on the above, the proof of Theorem 1 is derived as follows:

Proof. Based on Definition 1–3, we obtain the following specifications of g_m and g:

- In MV with m candidates, $\forall k > 0, g_m(0) = 1, g_m(k) = 0$, and $\forall x > 0, g(x) = 0, g(0) = 1$.
- In BV with m candidates, $\forall k \geq 0, g_m(k) = \frac{k}{m-1}$, and $\forall x > 0, g(x) = 1 - x$.

By Lemma 2, we obtain:

$$\mathbb{E}_{\mathcal{A}\sim\theta}\left[\frac{c(w_{MV}(\mathcal{A}))}{c(o(\mathcal{A}))}\right] \in \omega(1), \mathbb{E}_{\mathcal{A}\sim\theta}\left[\frac{c(w_{BV}(\mathcal{A}))}{c(o(\mathcal{A}))}\right] \in O(1),$$

and therefore Theorem 1 is proved. □

B Prompt Template for Estimating Preference

In this section, we present the prompt for estimating preferences, in which the question corresponds to the query, the context to the document d_i, and the answer to the candidate a_j.

Preference Estimation

```
<|begin_of_text|><|start_header_id|>system<|end_header_id|>
You are an expert fact-checking assistant. Your task is to
determine whether the given answer is factually supported
by the provided context.
You must answer "True" only if the answer is directly
supported by the context. If the answer is not explicitly
stated in the context you must answer "False".
Do not include any explanation. Output only one word:
"True" or "False".<|eot_id|>

<|start_header_id|>user<|end_header_id|>
Question: {question}

Context:
{context}

Answer:
{answer}
<|eot_id|><|start_header_id|>assistant<|end_header_id|>
```

GenAI Usage Disclosure

In accordance with the CIKM 2025 guidelines on responsible use of AI tools in research, we declare that no generative AI tools were used for the creation of code, datasets, or any novel content in this paper.

We solely employed generative AI tools (e.g., ChatGPT) for lightweight language editing and polishing, such as improving grammar, clarity, and fluency of the manuscript text. All scientific ideas, experimental designs, analyses, and written content were conceived and developed by the authors without assistance from generative AI systems.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774* (2023).

[2] Mistral AI. 2023. Mistral 7B Instruct. https://mistral.ai/news/announcing-mistral-7b. Accessed: 2024-05-13.

[3] Michel Balinski and Rida Laraki. 2020. Majority judgment vs. majority rule. *Social Choice and Welfare* 54, 2 (2020), 429–461.

[4] JC de Borda. 1781. M'emoire sur les' elections au scrutin. *Histoire de l'Acad'emie Royale des Sciences* (1781).

[5] Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. 2022. Improving language models by retrieving from trillions of tokens. In *International conference on machine learning*. PMLR, 2206–2240.

[6] Steven J Brams and Peter C Fishburn. 2002. Voting procedures. *Handbook of social choice and welfare* 1 (2002), 173–236.

[7] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems (NIPS '20)*. Article 159, 25 pages.

[8] Hung-Ting Chen, Michael Zhang, and Eunsol Choi. 2022. Rich Knowledge Sources Bring Complex Knowledge Conflicts: Recalibrating Models to Reflect Conflicting Evidence. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. 2292–2307.

[9] Xinyun Chen, Renat Aksitov, Uri Alon, Jie Ren, Kefan Xiao, Pengcheng Yin, Sushant Prakash, Charles Sutton, Xuezhi Wang, and Denny Zhou. 2023. Universal self-consistency for large language model generation. *arXiv preprint arXiv:2311.17311* (2023).

[10] Yu Cheng, Shaddin Dughmi, and David Kempe. 2018. On the distortion of voting with multiple representative candidates. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32.

[11] Sunhao Dai, Chen Xu, Shicheng Xu, Liang Pang, Zhenhua Dong, and Jun Xu. 2024. Bias and Unfairness in Information Retrieval Systems: New Challenges in the LLM Era. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '24)*. 6437–6447.

[12] Edith Elkind, Piotr Faliszewski, and Arkadii Slinko. 2010. Cloning in elections. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 24. 768–773.

[13] Edith Elkind, Piotr Faliszewski, and Arkadii Slinko. 2012. Clone structures in voters' preferences. In *Proceedings of the 13th ACM conference on electronic commerce*. 496–513.

[14] Ratip Emin Berker, Sílvia Casacuberta, Isaac Robinson, Christopher Ong, Vincent Conitzer, and Edith Elkind. 2025. From Independence of Clones to Composition Consistency: A Hierarchy of Barriers to Strategic Nomination. *arXiv e-prints* (2025), arXiv–2502.

[15] William V Gehrlein and Dominique Lepelley. 2010. *Voting paradoxes and group coherence: the Condorcet efficiency of voting rules*. Springer Science & Business Media.

[16] Google. n.d.. Google Search. https://www.google.com. Accessed: 2025-05-18.

[17] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783* (2024).

[18] Giwon Hong, Jeonghwan Kim, Junmo Kang, Sung-Hyon Myaeng, and Joyce Whang. 2024. Why So Gullible? Enhancing the Robustness of Retrieval-Augmented Models against Counterfactual Noise. In *Findings of the Association for Computational Linguistics: NAACL 2024*. 2474–2495.

[19] Cheng Hsu, Cheng-Te Li, Diego Saez-Trumper, and Yi-Zhan Hsu. 2021. WikiContradiction: Detecting self-contradiction articles on wikipedia. In *2021 IEEE international conference on big data (Big Data)*. IEEE, 427–436.

[20] David Ríos Insua and Simon French. 2010. *E-democracy: a group decision and negotiation perspective*. Vol. 5. Springer Science & Business Media.

[21] Zhuoran Jin, Pengfei Cao, Yubo Chen, Kang Liu, Xiaojian Jiang, Jiexin Xu, Li Qiuxia, and Jun Zhao. 2024. Tug-of-War between Knowledge: Exploring and Resolving Knowledge Conflicts in Retrieval-Augmented Language Models. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*. 16867–16878.

[22] Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*.

[23] Jaehyung Kim, Jaehyun Nam, Sangwoo Mo, Jongjin Park, Sang-Woo Lee, Minjoon Seo, Jung-Woo Ha, and Jinwoo Shin. 2024. Sure: Summarizing retrievals using answer candidates for open-domain qa of llms. *arXiv preprint arXiv:2404.13081* (2024).

[24] Kiseung Kim and Jay-Yoon Lee. 2024. RE-RAG: Improving Open-Domain QA Performance and Interpretability with Relevance Estimator in Retrieval-Augmented Generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. 22149–22161.

[25] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural Questions: A Benchmark for Question Answering Research. In *Transactions of the Association for Computational Linguistics*.

[26] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Yu, Joey Gonzalez, Hao Zhang, and Ion Stoica. 2023. vllm: Easy, fast, and cheap llm serving with pagedattention. *See https://vllm. ai/(accessed 9 August 2023)* (2023).

[27] Arian Mansouri, Pratyusha Varma, Hangfeng Zhao, Matt Gardner, Dan Roth, and Daniel Khashabi. 2023. PopQA: Open-Domain QA for Popular Entities. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

[28] Luigi Mittone. 2003. The Condorcet paradox: an experimental approach to a voting process.

[29] Liangming Pan, Wenhu Chen, Min-Yen Kan, and William Yang Wang. 2023. Attacking Open-domain Question Answering by Injecting Misinformation. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*. 525–539.

[30] Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language Models as Knowledge Bases?. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 2463–2473.

[31] Quang Hieu Pham, Hoang Ngo, Anh Tuan Luu, and Dat Quoc Nguyen. 2024. Who's Who: Large Language Models Meet Knowledge Conflicts in Practice. In *Findings of the Association for Computational Linguistics: EMNLP 2024*. 10142–10151.

[32] Ariel D. Procaccia and Jeffrey S. Rosenschein. 2006. The Distortion of Cardinal Preferences in Voting. In *Cooperative Information Agents X*. 317–331.

[33] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.* 21, 1, Article 140 (Jan. 2020), 67 pages.

[34] William H Riker and Peter C Ordeshook. 1968. A Theory of the Calculus of Voting. *American political science review* 62, 1 (1968), 25–42.

[35] Donald G Saari. 2023. Selecting a voting method: the case for the Borda count. *Constitutional Political Economy* 34, 3 (2023), 357–366.

[36] Pasi Shailendra, Rudra Chandra Ghosh, Rajdeep Kumar, and Nitin Sharma. 2024. Survey of Large Language Models for Answering Questions Across Various Fields. In *2024 10th International Conference on Advanced Computing and Communication Systems (ICACCS)*, Vol. 1. 520–527.

[37] Alan D. Taylor. 2005. *Social Choice and the Mathematics of Manipulation*. Cambridge University Press.

[38] Qwen Team. 2024. Qwen2: The Next Generation of Qwen Language Models. https://huggingface.co/Qwen. Alibaba Cloud.

[39] T Nicolaus Tideman. 1987. Independence of clones as a criterion for voting rules. *Social Choice and Welfare* 4, 3 (1987), 185–206.

[40] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and Efficient Foundation Language Models. (02 2023).

[41] Adrian Van Deemen. 2014. On the empirical relevance of Condorcet's paradox. *Public choice* 158 (2014), 311–330.
[42] Max Visser. 1998. *Five Theories of Voting Action: Strategy and Structure of Psychological Explanation.* Ph. D. Dissertation. University of Twente.
[43] Fei Wang, Xingchen Wan, Ruoxi Sun, Jiefeng Chen, and Sercan Ö Arık. 2024. Astute rag: Overcoming imperfect retrieval augmentation and knowledge conflicts for large language models. *arXiv preprint arXiv:2410.07176* (2024).
[44] Han Wang, Archiki Prasad, Elias Stengel-Eskin, and Mohit Bansal. 2025. Retrieval-Augmented Generation with Conflicting Evidence. *arXiv preprint arXiv:2504.13079* (2025).
[45] Zhepei Wei, Wei-Lin Chen, and Yu Meng. 2025. InstructRAG: Instructing Retrieval-Augmented Generation via Self-Synthesized Rationales. In *Proceedings of the International Conference on Learning Representations (ICLR)*.
[46] Orion Weller, Aleem Khan, Nathaniel Weir, Dawn Lawrie, and Benjamin Van Durme. 2024. Defending Against Disinformation Attacks in Open-Domain Question Answering. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 2: Short Papers)*. 402–417.
[47] Gwenyth Portillo Wightman, Alexandra Delucia, and Mark Dredze. 2023. Strength in numbers: Estimating confidence of large language models by prompt agreement. In *Proceedings of the 3rd Workshop on Trustworthy Natural Language Processing (TrustNLP 2023)*. 326–362.
[48] Marshall D Willman. 2022. Stanford Encyclopedia of Philosophy.
[49] Chong Xiang, Tong Wu, Zexuan Zhong, David Wagner, Danqi Chen, and Prateek Mittal. 2024. Certifiably robust rag against retrieval corruption. *arXiv preprint arXiv:2405.15556* (2024).
[50] Chen Xu, Wenjie Wang, Yuxin Li, Liang Pang, Jun Xu, and Tat-Seng Chua. 2024. A Study of Implicit Ranking Unfairness in Large Language Models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*. 7957–7970.
[51] Rongwu Xu, Zehan Qi, Zhijiang Guo, Cunxiang Wang, Hongru Wang, Yue Zhang, and Wei Xu. 2024. Knowledge Conflicts for LLMs: A Survey. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. 8541–8565.
[52] H Peyton Young. 1988. Condorcet's theory of voting. *American Political science review* 82, 4 (1988), 1231–1244.
[53] Wenhao Yu, Dan Iter, Shuohang Wang, Yichong Xu, Mingxuan Ju, Soumya Sanyal, Chenguang Zhu, Michael Zeng, and Meng Jiang. 2023. Generate rather than Retrieve: Large Language Models are Strong Context Generators. In *Proceedings of the International Conference on Learning Representations (ICLR)*.
[54] Yue Yu, Wei Ping, Zihan Liu, Boxin Wang, Jiaxuan You, Chao Zhang, Mohammad Shoeybi, and Bryan Catanzaro. 2024. RankRAG: Unifying Context Ranking with Retrieval-Augmented Generation in LLMs. In *Advances in Neural Information Processing Systems*, Vol. 37. 121156–121184.
[55] Qin Zhang, Shangsi Chen, Dongkuan Xu, Qingqing Cao, Xiaojun Chen, Trevor Cohn, and Meng Fang. 2023. A Survey for Efficient Open Domain Question Answering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 14447–14465.
[56] Fengbin Zhu, Wenqiang Lei, Chao Wang, Jianming Zheng, Soujanya Poria, and Tat-Seng Chua. 2021. Retrieving and reading: A comprehensive survey on open-domain question answering. *arXiv preprint arXiv:2101.00774* (2021).