Enabling Roll-up and Drill-down Operations in News Exploration via Knowledge Graphs

No Author Given

No Institute Given

Abstract. The task of effective news exploration is often challenging for journalists and analysts. This paper introduces NCExplorer, a novel framework that incorporates OLAP-like operations to streamline the news exploration process. Key features include *roll-up* operations, which enable users to select preferred concepts for examining similar content within a broader context, and *drill-down* operations, which offer subtopic suggestions for more focused exploration. These operations are achieved by harnessing the power of external knowledge graphs (KGs) through both fact and ontology networks. A unique ranking scheme is proposed for linking news articles with KG concepts, and an efficient, unbiased sampler is developed to scale rank computation for large KGs, overcoming the limitations of graph path enumerations. Experimental results from AI evaluators and crowd-sourced participants on Amazon Mechanical Turk demonstrate that NCExplorer is the preferred tool for news exploration across various topic domains, outperforming state-of-the-art news search approaches on real-world news datasets.

1 Introduction

Effectively exploring a vast news corpus is vital for numerous analytical tasks, particularly for professional users. For instance, journalists need to gather relevant news articles to create in-depth analyses for feature stories or editorials [35,12]. Similarly, traders depend on real-time updates and historical reports to mitigate risks in fluctuating markets [24,30]. This highlights the importance of efficient and comprehensive news exploration tools for various professional applications. The analytical requirements for structured data have been effectively addressed by OLAP processing methods, which allow users to iteratively perform *roll-up* operations for a bird's-eye view or *drill-down* operations to investigate details further. The value and applicability of this interactive flow extend beyond structured data, making it equally useful for unstructured data and an essential tool for analysis and insight generation in today's information-rich world.

Suppose analyst Alice discovers a news article about "*Tesla Motors*" and wishes to delve deeper into the topic. To do this effectively, Alice needs to explore not only news related to Tesla but also its competitive landscape, including other electric vehicle players like "*Waymo*" and "*NIO*" as well as traditional car manufacturers. This requirement resembles a *roll-up* operation from "*Tesla*" to car makers. Additionally, Alice wishes

Table 1: NCExplorer ***roll-up*** example

News Article	Lukoil, a Russian oil company, calls for an end to the Ukraine war.
Document Entities	Russian, Ukraine, war
Rollup concepts	European countries, Conflict
Implicitly matched documents	1. U.S. weighs Russian oil ban as gas prices surge and Ukraine war grows. 2. NATO accuses Russia of using cluster bombs in Ukraine. 3. NATO rejects intervening in Ukraine to avoid the risk of a wider European war. 4. Boeing and Ford suspend operations in Russia as the country escalated its war in Ukraine. 5. What Putin's war could mean for fossil fuels. 6. France: Government Introduces Terrorism Legislation. 7. Don't limit Ukraine Refugee numbers – Sturgeon

to examine the varying development statuses of electric vehicles across geographic regions, such as "*Europe*", "*North America*", and "*Asia*". In this context, a *drill-down* operation is necessary to narrow down the news content and uncover market insights. Although many commercial search engines currently provide navigation features by suggesting phrases that people frequently search for, these recommendations are primarily influenced by crowd behaviors rather than the underlying ontology within the corpus. As a result, there is a need for more sophisticated search and exploration tools that can effectively uncover relevant content based on the users' specific interests and analytical requirements, rather than just relying on popular search trends.

To address these challenges, we introduce NCExplorer, an innovative framework for systematic and insightful news exploration. Given a news article, NCExplorer generates semantic *roll-up* and *drill-down* operations based on the detected entities to support user exploration. For instance, in Table 1, a news article comprises two entities, "*Ukraine*" and "*War*". For each entity, NCExplorer generates related concepts for *roll-up* operations. For example, "*Ukraine*" can be generalized to "*European Countries*" or an even broader concept like "*UN countries*". Once users select the rolled-up concepts, such as generalizing ("*Ukraine*", "*War*") to ("*European Countries*", "*Conflict*""), NCExplorer returns relevant news articles and highlights the connections between each result and the chosen rolled-up concepts. Furthermore, as illustrated in Fig. 1,NCExplorer identifies novel subtopics from the related news articles for the *drill-down* operation.

For instance, among the news articles corresponding to ("*European Countries*", "*Conflict*"), NCExplorer identifies "*Natural Gas*", "*Oil*", and "*NATO*" as subtopics related to these conflicts. This added layer of granularity allows users to delve into particular aspects of their selected topic and obtain more profound insights, leading to a more comprehensive and

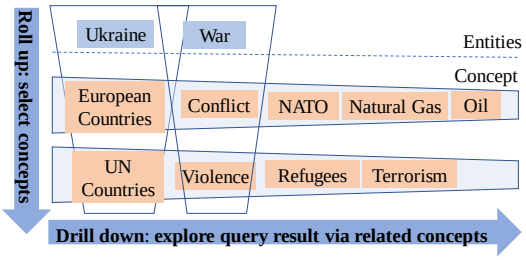

Fig. 1: NCExplorer ***drill-down*** example

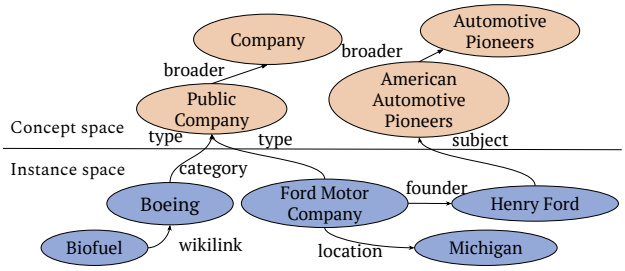

Fig. 2: KG Concept and Instance Spaces

focused analysis. To facilitate the *roll-up* and *drill-down* operations, NCExplorer relies on external knowledge graphs (KGs) [34,19]. These KGs comprise not only millions of entities and relationships forming a fact network but also rich ontology information. However, there are two major challenges that need to be addressed when building NCExplorer.

The first challenge involves effectively computing the relevance between a concept and a news article. For example, consider the fourth document in Tab. 1, "*Boeing and Ford suspend operations in Russia*" [25]. The entity "*Ford Motor Company*" matches several concepts based on the KG ontology, such as "*Multinational Companies Headquartered in the United States*" and "*Motor Vehicle Manufacturers*". However, "*Multinational Companies Headquartered in the United States*" is more relevant because it also generalizes another entity in the news: "*Boeing*". With this observation in mind, we design a novel ranking function that considers both 1) *ontology relevance* and 2) *context relevance*. Ontology relevance deems a concept relevant to a news document if the linked entity is essential to the document. As for context relevance, a concept's weight is determined by its relationship with the remaining entities (also known as context entities) in the news. However, the associations between a concept c and the context entities in the news are not captured by the KG ontology. To address this issue, we leverage the fact network in the KGs and evaluate contextual relevance through the paths in the fact network connecting entities categorized by c with all context entities. While effective, computing the contextual relevance is prohibitively expensive, as it requires path enumerations between numerous entity pairs. To overcome this limitation, we scale NCExplorer by proposing an efficient random walk sampling algorithm that provides an unbiased estimation of the exact relevance score.

The second challenge lies in ranking concepts for the NCExplorer operation. Since the number of concepts generated from a query result can be large (e.g., 956 subtopics are detected on news related to "Corporate Finance"), we develop an effective ranking scheme that incorporates three key factors: *coverage*, *specificity*, and *diversity*. First, a desired subtopic should cover a sufficient number of news documents because topics with limited coverage are less critical for analytical tasks. Second, our ranking scheme prioritizes concepts with higher specificity (fewer neighbors

in the KG) to penalize broad concepts like "*Living Person*" and "*Organization*". Third, to reduce media coverage bias where a small number of popular entities receive the most news visibility, we prioritize higher diversity, ensuring that more distinct entities can be found in the *drill-down* result.

The main contributions of our work can be summarized as follows::

- We introduce a novel framework, NCExplorer, which is the first of its kind, supporting news exploration with semantic *roll-up* and *drill-down* operations by leveraging the ontology and fact networks of knowledge graphs (KGs) (Sec. 3)
- We develop effective ranking schemes to evaluate the relevance between concepts and news articles and devise an efficient, unbiased sampling estimator for relevance score computation. (Sec. 4)
- We evaluate NCExplorer on real-world news datasets with crowdsourced participants from Amazon Mechanical Turk. Experimental results, with more than 3,900 ratings from master-qualified participants and 1200 ratings from chat-GPT, confirm that NCExplorer's ranking schemes are more effective than those of state-of-the-art news search methods. Additionally, through a case study, NCExplorer demonstrates its exploration capability in identifying media bias. Our demo system is available at https://ncexplorer.knowledge-fusion.science/ (Sec. 5)

2 Related Work

Our work is broadly related to news search and recommendation, aiming to enhance user experience in these domains. Existing works in this area typically extend generic natural language understanding (NLU) models [10,29,32] to improve search and recommendation experiences for news articles. These works generally fall into two categories: (a) embedding-based approaches and (b) structure-based approaches.

Embedding-based approaches: These methods leverage embeddings to create document representations that capture the semantic meaning of news articles. Notable works in this category include DKN [36], KRED [23], and NewsGraph [21]. DKN integrates the embeddings of news titles, derived from a CNN approach [18], with the embeddings of news KG entities obtained through a knowledge graph embedding technique [17]. KRED extends this idea further by introducing a context embedding layer generated from a news entity's neighbors, as well as an attentive merging layer inspired by the Knowledge Graph Attention Network (KGAT). NewsGraph, on the other hand, builds a separate graph by removing irrelevant edges and introducing new edges that represent co-occurrences in the same news or visits from the same user. This alteration hydrates the static KG with dynamic interactions between entities in real life, making the latent vectors aware of current affairs.

Structure-based approaches: These methods focus on utilizing the structure of KGs to provide better explainability and semantic relevance between news articles. AnchorKG [22] and NewsLink [37] are two prominent works in this category. AnchorKG represents each news article as

a compact subgraph containing essential news entities and their k-hop neighbors, generated using a reinforced learning framework. This interaction of anchor graphs provides explanations for semantic relevance between news articles. NewsLink, a state-of-the-art news search approach, uses seed nodes identified from text fragments and a graph expansion algorithm to connect nodes in a single graph, adding hidden related nodes as auxiliary information for news semantic representation.
Our work is more closely related to structure-based approaches, as these methods offer better explainability and enable exploration. We use news entities as the starting point and follow their ontology paths to generate high-level concepts for news articles. This approach differentiates NCExplorer from existing works, as it focuses on the entities' ontology path instead of looking among neighbors for additional context information. In summary, NCExplorer is a novel approach that leverages the strengths of structure-based methods to provide a more comprehensive understanding of news content. To the best of our knowledge, our work is the first to facilitate systematic exploration with OLAP-like operations.

3 An Overview of NCExplorer

NCExplorer leverages external KGs for news exploration. A KG is a multigraph $\mathcal{G} = (\mathcal{V}_C \bigcup \mathcal{V}_I, \mathcal{E}_C \bigcup \mathcal{E}_I, \Psi)$. $\mathcal{V}_C$ and $\mathcal{V}_I$ represent the concept entities and the instance entities respectively. Each concept edge in $\mathcal{E}_C$ links two concept entities, and an instance edge in $\mathcal{E}_I$ links two instance entities. Like [37], we add a reversed edge for each original edge so that $\mathcal{G}$ is bidirected. The associations between the instance space and the concept space are captured by the ontology relation Ψ. $\Psi(c)$ maps a concept entity $c \in \mathcal{V}_C$ to a set of instance entities while $\Psi^{-1}(v)$ maps an instance entity $v \in \mathcal{V}_I$ to a set of concept entities.

Example 1. Figure 2 shows a KG example from wikidata. "*Boeing*", "*Biofuel*", and "*Henry Ford*" are instance entities while "*Public Company*" and "*American Automotive Pioneers*" are concept entities. The ontology relation Ψ maps "*Public Company*" to two instance entities whereas Ψ^{-1} maps "*Henry Ford*" to two concept entities. "*Public Company*" has only one neighbor in the concept space, and "*Ford Motor Company*" has only two neighbors in the instance space.

There are two major components in the architecture of NCExplorer: *news indexing*, semantic *roll-up*, and *drill-down*:

News Indexing. Each news document $d \in \mathcal{D}$ is prepossessed by standard NLP techniques. We build a linker using the Spacy library [15]. The end product is a set of instance entities extracted from d. For each instance entity $v \in d$, we map v to the set of concept entities by the KG ontology relation, i.e., $\Psi^{-1}(v)$. Subsequently, we take any concept entity $c \in \Psi^{-1}(v)$ and create an inverted index entry $< c, d, cdr(c, d) >$ where $cdr(c, d)$ is the relevance score of concept entity c on document d. To ease the presentation, we use entity to denote *instance entity* and concept to denote *concept entity*.

Note that the accuracy of entity linking affects the quality of the extracted concepts for a news document. Although improving entity linking accuracy is not the focus of this work, the design of the ranking function $cdr(c, d)$ penalizes irrelevant concepts based on our proposed *ontology relevance* and *context relevance*. The details will be discussed in Sec. 4.1.

Roll-up Operation. After NCExplorer generates a list of entities from a news document, users then replace one or more entities with KG concepts to form a concept pattern query Q. Given such a Q, a document d matches Q if for each concept $c \in Q$, there is an entity v in d that $v \in \Psi(c)$. In other words, we can find a set of entities in d that match Q. The relevance score of a matched document d to Q is the sum of the relevance scores among all concepts in Q, i.e.,

$$rel(Q, d) = \sum_{c \in Q} cdr(c, d). \tag{1}$$

NCExplorer returns all matched documents to the query Q, ranked by the relevance scores $rel(Q, \cdot)$. Furthermore, the entities that match the concepts in Q are annotated to explain why a news document is retrieved.

Drill-down Operation. Based on the matched news from the *roll-up* operation, NCExplorer suggests additional concepts as subtopics to a query Q. A suggested concept c', as a subtopic of Q, enables users to conduct a *drill-down* analysis. By selecting c', users narrow down the matched news to the augmented query $Q \cup c'$. Since a subtopic must appear as a concept in at least one matched document of Q, we can find all candidate subtopics by unionizing the concepts from all matched documents. For efficient query processing, we construct a forward index by storing the concept set of each document, i.e., $< d, c >$. Since there could be a large number of concepts as subtopics, we design an effective ranking scheme and only return top-scored subtopics. The design of the ranking scheme will be discussed in Sec. 4.2.

4 News Ranking & Exploration

In this section, we first discuss the ranking schemes in NCExplorer: Section 4.1 presents the relevance score between a concept c and a document d, i.e., $cdr(c, d)$; Section 4.2 presents the subtopic score for a concept c and a query Q, i.e., $sbr(c, Q)$. In Section 4.3 we discuss how to efficiently compute the scores with sampling techniques.

4.1 Concept Document Rank

We propose a novel ranking scheme for $cdr(c, d)$ by considering two key relevance dimensions: *ontology relevance* $cdr_o(c, d)$ and *context relevance* $cdr_c(c, d)$. A concept c is relevant to a document d if c is relevant in both dimensions:

$$cdr(c, d) = cdr_o(c, d) \cdot cdr_c(c, d) \tag{2}$$

Ontology Relevance. When a document d contains an entity v that matches a concept c under the ontology relation Ψ, i.e., $v \in \Psi(c)$, c is associated with d by ontology relevance. Since there could be more than one entity in d that matches c, we define the match entity set for c and d as follows:

Definition 1 (Match Entity Set). *Given a concept c and a document d, $ME(c,d)$ is the set of entities that are found in d and match c, i.e., $ME(c,d) = \{v|\ v \in d \text{ and } v \in \Psi(c)\}$.*

We now define the ontology relevance score function as follows:

$$cdr_o(c,d) = log\frac{|\mathcal{V}_I|}{|\Psi(c)|} \cdot \left(\max_{v \in ME(c,d)} tw(v,d)\right) \tag{3}$$

First, a concept that matches more entities in the ontology, i.e., lower specificity, should be less relevant to a document. Second, among all the matched entities, a *pivot* entity is selected as the one with the highest term weight $tw(v,d)$ in the document to match the concept. The term weight is used to capture the importance of v in d. If v plays a more significant role in d, then c is more relevant to d. We use the typical TF-IDF scheme for the term weight in our implementation and other schemes can be easily supported as well.

Context Relevance. The ontology relevance associates a concept with a document by the ontology relation Ψ directly. However, an entity in a document could map to different concepts by the ontology. Therefore, ontology relevance can only differentiate the matched concepts of the same entity with the specificity score $\log(|\mathcal{V}_I|/|\Psi(c)|)$, which simply penalizes broad concepts. We thus propose to include the unmatched entities in the document as contextual information to improve the relevance semantics.

Definition 2 (Context Entity Set). *Given a concept c and a document d, $CE(c,d)$ is the set of entities found in d and the entities do not match c, i.e., $CE(c,d) = \{v|\ v \in d \text{ and } v \notin \Psi(c)\}$.*

To measure the context relevance of c and d, we compute the relevance of c and the context entities $CE(c,d)$. Although the context entities do not match c, we take advantage of both the ontology relation and the instance space to link c with the context entities. We introduce a novel connectivity score $conn(c,d)$ to measure the KG connectivity between a concept c and a context entity set $CE(c,d)$ as follows:

$$conn(c,d) = \sum_{v \in CE(c,d)} \frac{\sum_{u \in \Psi(c)} \sum_{l=1}^{\tau} \beta^l \cdot |paths_{u,v}^{<l>}|}{|CE(c,d)|} \tag{4}$$

where $|paths_{u,v}^{<l>}|$ measures the number of l-hop simple paths connecting u and v in the instance space, and β is a damping factor that penalizes longer paths.

Intuitively, the connectivity score is the average number of paths among all context entities connected to any KG instance entity that matches c,

subject to a hop constraint of at most τ. Thus, a better connectivity leads to a higher relevance score. Unlike existing news search approaches that incorporate KGs [11,37], we do not assume the entities in a document are connected. Finally, we normalize the connectivity score to $[0, 1)$ as follows:

$$cdr_c(c, d) = 1 - \frac{1}{1 + conn(c, d)} \tag{5}$$

4.2 Drill-down Exploration Rank

A large number of relevant documents can be retrieved for a concept pattern query Q with the roll-up operation. To facilitate users in navigating the results with a concept-centric approach, NCExplorer automatically detects related concepts in the retrieved documents as the suggested subtopics for users to perform *drill-down* analysis.

Let $\mathcal{D}(Q)$ denote the set of retrieved documents for Q and a candidate subtopic is a concept c such that there is an instance entity $v \in \Psi(c)$ appearing in one of the documents $d \in \mathcal{D}(Q)$. For example, "*Multinational Companies with HQ in the U.S*" is a candidate subtopic for the query in Figure 1 since "*Boeing*" and "*Ford Motor Company*" are matches of the subtopic, and they appear in the news retrieved for the query. We develop an effective subtopic score $sbr(\cdot, Q)$ over candidate subtopics and only suggest top-scored subtopics for Q. There are three key perspectives considered in $sbr(\cdot, Q)$ as below:

$$sbr(c, Q) = coverage(c, Q) \cdot specificity(c) \cdot diversity(c, Q) \tag{6}$$

- **Coverage.** Among all the documents retrieved for Q, a niche subtopic that is only related to a small number is less attractive to users for exploration. Thus, we use the total relevance of a concept to all the documents in $\mathcal{D}(Q)$ as the coverage score, i.e., $coverage(c, Q) = \sum_{d \in \mathcal{D}(Q)} cdr(c, d)$.
- **Specificity.** Similar to the ontology relevance, a broad concept that matches a large number of entities is less desirable as it could lead to suggesting a trivial subtopic, such as "*Person*". Hence, we use $specificity(c) = \log(|\mathcal{V}_I|/|\Psi(c)|)$ to prioritize concepts with higher specificity scores.
- **Diversity.** We observe that some popular entities frequently appear in news documents. If not considering diversity, the subtopic score is biased towards the concepts that match these popular entities. To mitigate such a bias, we design the diversity score to favor concepts that match more distinct entities i.e.,

$$diversity(c, Q) = \frac{|\bigcup_{d \in \mathcal{D}(Q)} ME(c, d)|}{|\mathcal{D}(Q \cup \{c\})|} \tag{7}$$

  where we normalize the score as the average number of distinct entities mapped to the concept c among all the documents relevant to $Q \cup \{c\}$. This is to ensure the fairness of suggested concepts. Otherwise, the results can easily bias towards those concepts that match a small set of popular entities appearing in a large number of documents.

4.3 Connectivity Score Estimation

Although the connectivity score can effectively detect relevant concepts for both news matching and news exploration purposes, the score is prohibitively expensive to compute for two reasons. First, consider the rightmost summation of Equation 4, it requires to enumerate all simple paths up to τ-hop that connect a context entity with any KG instance entity mapped to concept c. The path enumeration of each entity pair is expensive as existing studies on s-t path enumeration can only produce *polynomial delay* algorithms such that the time between finding two successive path results is bounded by a polynomial function of the input size in the worst case [28,33]. Second, there is a massive number of s-t path enumeration tasks to execute for just processing one news document in the indexing component of NCExplorer. For each ontology-relevant concept c, we need to compute the context relevance score between c and each context entity in d for creating the index entry $< c, d, cdr(c, d) >$. Furthermore, the number of instance entities that map to c in the KG is massive, especially for broad concepts like "*Company*" and "*People*".
Inspired by the sampling approaches for online join aggregation queries [20,38,26], we devise a single random walk estimator to reduce the expensive cost of computing the context relevance scores. To start a random walk, we first sample a source entity u from all entities that map to the concept c, and a target entity v from the corresponding context entities. A *non-repeating* random walk $r(u, v)$ from u tries to reach v samples via at most τ distinct nodes. Let u_i denote the ith node sampled by a random walk starting from the source u and $N(u_i)$ is the number of eligible neighbors of u_i to be sampled, we define the following sample estimator:

$$r(u, v) = \mathcal{I}(u, v) \cdot |\Psi(c)| \cdot \beta^{l-1} \cdot \prod_{i=1}^{l-1} N(u_i) \tag{8}$$

where $\mathcal{I}(u, v)$ is an indicator random walk: it returns 1 if u reaches v at the l-th sampled node for any $l \leq \tau$; otherwise, it returns 0.
We run multiple random walks and use the sample mean to approximate the connectivity score, i.e., $\widehat{conn}(c, d) = \frac{1}{\theta} \sum_{j=1}^{\theta} r_j$.
The following theorem ensures our sample approach is unbiased.

Theorem 1. *$r(\cdot, \cdot)$ is an unbiased estimator to $conn(c, d)$, i.e., $\mathbb{E}[r(\cdot, \cdot)] = conn(c, d)$ where the randomness is taken over the source u and target v as well as the random walk from u to v.*

Proof Sketch. $conn(c, d)$ is a weighted sum of all the paths connecting an entity node $u \in \Psi(c)$ and a context entity $v \in CE(c, d)$ subject to a hop constraint of τ. For each path $s = \{u_1 = u, u_2, u_3, ..., u_l = v\}$ that connects u and v, the probability of s being sampled is $\mathbb{P}(s) = (|\Psi(c)| \cdot |CE(c, d)| \cdot \prod_i^{l-1} N(s_i))^{-1}$. Thus, we can use the Horvitz–Thompson estimator [16] to approximate the population sum without bias.

Convergence Optimization. Although we devise an unbiased estimator for the connectivity score, the sampling process may suffer from slow convergence, especially when many of the sampled paths cannot reach the target context entity v. To enable faster convergence, we build a

Algorithm 1: Random walk estimator with k-hop index.

Input : KG $\mathcal{G}$, concept c, document d, sample size θ
Output: the estimated connectivity est

1 $est \leftarrow 0$;
2 **while** *there are less than* θ *random walks* **do**
3 $l \leftarrow 1$; $p \leftarrow 1.0$;
4 $u_l \leftarrow$ a random entity in $\Psi(c)$;
5 $v \leftarrow$ a random entity in $CE(c, d)$;
6 **while** $l \leq \tau$ **do**
7 $count \leftarrow 0$;
8 **foreach** $n \in N(u_l)$ **do**
9 **if** $hop(n, v) \leq \tau - l$ **then**
10 $count \leftarrow count + 1$;
11 $u_{l+1} \leftarrow n$ with probability $\frac{1}{count}$;
12 $p \leftarrow \frac{p}{count}$;
13 $l \leftarrow l + 1$;
14 **break** on $u_l = v$;
15 **if** $u_l = v$ **then**
16 $est \leftarrow est + \frac{\beta^{l-1}}{p}$;
17 **return** $\frac{est \cdot |\Psi(c)|}{\theta}$.

reachability index [9] on the KG instance space and only sample eligible neighbors that satisfy the hop constraint. We present our sampling approach with the k-hop reachability index in Algorithm 1. For each random walk, we first sample the source u_1 and the target v. Subsequently, we iteratively sample the next node u_{l+1} by scanning the neighbors of u_l. With the help of the k-hop index, we first check if a neighbor n can reach v within the hop constraint of $\tau - l$ (Line 9) Then, we uniformly sample eligible neighbors by a Reservoir sampler, which only requires a single scan of the neighbors (Line 11) A random walk terminates when either the target v is sampled or the hop limit is reached.

5 Evaluation

We present the setup in Sec. 5.1 and then answer the following questions: (1) Can NCExplorer produce relevant results given topics rolled up from news articles (Sec. 5.2)? (2) Can *context relevance score* effectively measure the relevance between a concept and a document? How does our sampling method impact the accuracy (Sec. 5.4)? (3) How effective is each scoring component in the ranking model to the drill-down operation (Sec. 5.5)? (4) Can NCExplorer support real analytical applications (Sec. 5.6)?

5.1 Settings

Datasets. We use DBPedia [1] as our backend KG. Only the English version is used in experiments but NCExplorer is not language specific. The June 2021 snapshot contains 1 billion triples with about 1.6 million people, 1 million places, and 0.3 million organizations. We crawl 200k news articles from popular news portals: *Reuters* [4], *Seeking Alpha* [5] and *The New York Times* [2] to have a mixture of business and politics reports. The dataset is released together with the source code [1].

Compared Methods.

- Lucene implements a typical bag-of-words keyword match model. We use BM25 [31] for the term weighting scheme with the default library settings. We use Lucene's subproject Solr [13] for the experiment. Both Lucene's and Solr's versions are 8.11.1.
- Bert [10] is a popular neural text embedding model. We use a SBERT [6], a modification of the pre-trained Bert to map each news article to a vector of 768 dimensions. We use the vector search engine Qdrant [7] for embedding storage and query.
- Newslink [37] is the state-of-the-art implicit news exploration method that expands a news document and a query by forming a common ancestor graph extracted from the KG. Each KG entity in the extracted graph is then treated as a matching keyword in the bag-of-words model. The default parameters are used, and the term weights are set by BM25.
- Newslink-Bert is a hybrid method that combines Newslink and Bert. We first expand the entities in a query into a subgraph using Newslink's algorithm. We then concatenate entities in the subgraph to form a long text query. The Bert embedding of the long text query is sent to Qdrant search engine for vector matching.
- NCExplorer is the proposed approach in this paper. The parameters are set to $\tau = 2$ and $\beta = 0.5$ by default. The number of samples for connectivity score estimation is set to 50.

Implementation. NCExplorer and Newslink are implemented in python 3.9. Bert and Newslink-Bert use Qdrant as vector search engine [7]. We use a server running on Ubuntu 20.04 with an AMD EPYC 7643 Processor @ 3.45GHz and 251G RAM.

5.2 Concept Document Relevance Study

We evaluate a total of six topics: *International Trade*, *Lawsuits*, *Elections*, *Mergers&Acquisitions*, *International Relations* and *Labor Dispute*. Each topic is combined with either a country or an industry to form queries such as *Elections in African countries* or *Lawsuits involving U.S. technology companies*. For each query, the top 5 news articles are retrieved from each method, resulting in 25 outcomes. These results are presented to each evaluator in a randomized order. To ensure a fair comparison, we hide NCExplorer's result explanations. The relevance level is rated for each concept in the query, with values ranging between 0 and 5. In

[1] https://anonymous.4open.science/r/ncexplorer-1F32/

	ndcg@1	ndcg@5	ndcg@10	ndcg@1	ndcg@5	ndcg@10
Topic	International Trade			Lawsuits		
Lucene	0.705	0.571	0.745	0.578	0.608	0.772
BERT	0.847	0.883	0.949	**0.837**	0.853	0.936
NewsLink	0.807	0.641	0.792	0.341	0.422	0.645
NewsLink-BERT	0.812	0.874	0.945	0.641	0.806	0.887
NCExplorer	**0.960**	**0.951**	**0.983**	0.833	**0.915**	**0.957**
Topic	Elections			Mergers & Acquisitions		
Lucene	0.557	0.459	0.656	0.472	0.601	0.784
BERT	0.878	0.894	0.942	**0.754**	**0.835**	0.924
NewsLink	0.553	0.465	0.660	0.322	0.469	0.693
NewsLink-BERT	**0.941**	**0.970**	**0.989**	0.742	0.809	0.917
NCExplorer	0.918	0.956	0.977	0.690	0.824	**0.928**
Topic	International Relations			Labor Dispute		
Lucene	**0.939**	0.757	0.850	0.565	0.630	0.820
BERT	0.914	0.917	0.954	0.398	0.441	0.687
NewsLink	0.777	0.760	0.852	0.478	0.489	0.721
NewsLink-BERT	0.893	0.956	0.977	0.720	0.742	0.897
NCExplorer	0.929	0.969	**0.984**	**0.913**	**0.979**	**0.987**

Table 2: NCDG@k from human evaluators. The best results are boldfaced and the second-best results are underlined.

total, we obtain 3,900 human ratings from 78 evaluators and 600 ratings from each of the two GPT evaluators. We use **NDCG@k** to evaluate the effectiveness of relevance ranking.

Human Evaluator: The results obtained from human evaluators are presented in Table 2. NCExplorer consistently achieves the best or second-best performance in all cases, except for the query "*Mergers & Acquisitions, U.S. biotechnology companies*". A detailed analysis reveals that evaluators show greater confidence in commonly known surface words like "*MA*", "*acquisitions*", "*buy*", and "*sell*", while expressing uncertainty about specialized terms such as "*takeover*". This discrepancy is not observed for GPT evaluators.

This finding highlights the effectiveness of NCExplorer's roll-up operation in helping analysts collect pertinent news articles featuring more domain-specific vocabularies. Embedding models also demonstrate remarkable performance in news exploration tasks, particularly when combined with KG information. However, there are two issues that arise when using pure embedding models. First, implicit matching may retrieve reports like daily trade price/volume[2], which, although crucial for real-time decision-making, have limited utility in exploration tasks. Second, the issue emerges when the concept entity isn't frequently mentioned alongside instance entities. For example, "*Labor disputes*" are often reported in the news as strikes organized by various labor unions. The embedding of "*Labor disputes*" alone cannot yield high-quality results. This issue can be

[2] example: https://www.bloomberg.com/markets/stocks/futures

	Human	**GPT-3.5**	**GPT-4**	**Human**	**GPT-3.5**	**GPT-4**
Topic	International Trade			Lawsuits		
Lucene	0.571	0.333	0.397	0.608	0.136	0.382
BERT	0.883	**0.950**	0.823	0.853	0.666	0.764
NewsLink	0.641	0.167	0.424	0.422	0.072	0.189
NewsLink-BERT	0.874	0.898	0.853	0.806	0.610	0.624
NCExplorer	**0.951**	0.903	**0.964**	**0.915**	**0.881**	**0.928**
Topic	Elections			Mergers & Acquisitions		
Lucene	0.459	0.309	0.392	0.601	0.429	0.422
BERT	0.894	0.761	0.891	**0.835**	0.686	0.538
NewsLink	0.465	0.330	0.254	0.469	0.006	0.126
NewsLink-BERT	**0.970**	**1.000**	**0.997**	0.809	**0.866**	0.644
NCExplorer	0.965	0.911	0.969	0.824	0.700	**0.977**
Topic	International Relations			Labor Dispute		
Lucene	0.757	0.333	0.397	0.630	0.496	0.502
BERT	0.917	0.950	0.823	0.441	0.509	0.153
NewsLink	0.760	0.167	0.424	0.489	0.186	0.357
NewsLink-BERT	0.956	**0.898**	0.853	0.742	0.598	0.519
NCExplorer	**0.969**	0.903	**0.964**	**0.979**	**0.959**	**0.998**

Table 3: NDCG@5 from human and GPT evaluators.The best results are boldfaced and the second-best results are underlined.

partially mitigated with supplementary context provided by Newslink's subgraph KG embedding.

When depending solely on Newslink, the performance stability is compromised due to the infrequent formation of densely connected single components by the subgraph embedding of multiple concept entities. Instead of identifying hidden nodes that connect existing query entities, the subgraph often results in a single concept entity accompanied by its N-hop neighbors. This dilutes the significance of concepts with smaller connected components, affecting the overall effectiveness of the news exploration process.

GPT Evaluators: GPT evaluation is conducted via OpenAI's chat completion API [3]. System role is set at "*You are a helpful assistant.*" The prompt template is "*<News Article>: Is this article related to <Query Concepts>? Please give a rating between 0 and 5, with 5 being the most relevant. Only give a one-decimal number. If cannot determine, return 0.*" An example question and rating are provided as chat history. The instance entities that appeared in the news article are supplied to help GPT understand specialized knowledge such as "*what are the biotechnology companies in the U.S.*" The sampling temperature is set to 0. The results of GPT evaluators are shown in Table 3. NDCG@5 is selected as the evaluation metric. Human evaluators' result is also listed as a reference. ratings from the GPT evaluators align with those made by hu-

[3] https://platform.openai.com/docs/api-reference/chat/create

mans. This observation is in line with the findings of another research [14] that compares ChatGPT with human evaluators on NLU-related tasks. GPT-4 [3], which possesses more advanced reasoning skills, gives NCExplorer the highest rating.

5.3 Efficiency Study

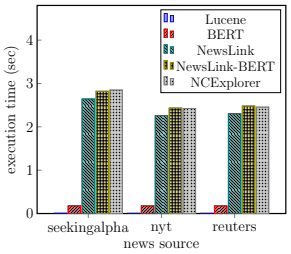

Fig. 3: Indexing time

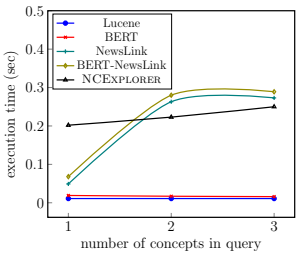

Fig. 4: Retrieval time

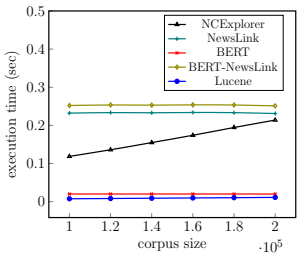

Fig. 5: Scalability Study

Indexing Efficiency. NCExplorer processes each news document and constructs an index for query processing. To evaluate the indexing overhead, we select 100 articles from each news portal and report the average processing time in Fig. 3. Lucene and Bert show sub-second execution time. Newslink, Newslink-Bert, and NCExplorer cost 2-3 seconds for each article. A breakdown analysis of the total cost showed that the top two overheads come from entity linking (91.8%) and relevance score calculation (7.1%). Entity linking is a common cost to all methods that require KG analysis. Improving the efficiency of entity linking is beyond the scope of this work. The relevance score calculation only takes 7.1% of the overall index time due to our efficient estimation approach via sampling. Furthermore, the overhead is manageable as the indexing cost incurs only once per document and the process can be easily accelerated with multi-threading or distributed approaches.

Retrieval Efficiency. Fig 4 displays the query efficiency of the compared methods when increasing the number of concepts while keeping corpus size fixed. The processing time for each data point is the average across 100 queries. Lucene is a highly optimized system for text retrieval and takes the least amount of time. Bert used to take longer execution time. Due to the absence of index, each query requires a comparison of all document embeddings to determine relevance. Recent development on vector databases has greatly sped up embedding retrieval and result in Lucene compatible speed. The performance for NCExplorer is similar to Newslink where the duration correlates to number of KG entities in the query. Newslink-Bert takes the sum of Bert and Newslink. Overall, NCExplorer can answer a concept pattern query with reasonable overhead.

Fig. 5 shows the relationship between retrieval time and corpus size for the set of queries. Lucene and Bert achieve the best scalability thanks to open source solutions. NCExplorer and Newslink have higher overhead because the queries are performed on a document database instead

of search engines. It is possible improve KG based methods by using Lucene to retrieve the matched documents before ranking the results with database records. The implementation is skipped in this study as the sub-second retrieval time does not degrades system usability.

5.4 Context Relevance Score

Effectiveness and Parameter Study. To verify the effectiveness of the context relevance score $cdr_c(c, d)$ (Eqn. 4), we design a "negative sampling" approach. We randomly select 100 entries from the inverted index $< c, d >$ where c is a relevant concept that links to at least one entity in d. For each entry $< c, d >$, we sample a concept node c' from the KG to generate a "*negative*" concept. Fig. 6 shows that the context relevance score can effectively differentiate c and c' as the relevant concepts have higher scores than those of negative concepts, i.e., $cdc_c(c, d) > cdc_c(c', d)$, regardless of the hop constraint τ from 1 to 3. Furthermore, when τ is set to 1 and 2, the score differences are significant compared with $\tau = 3$. It implies that setting a large τ can include irrelevant concepts as there is a higher chance to link concepts with documents. We use $\tau = 2$ as the default as more than half (55%) of the relevant scores are 0 when $\tau = 1$. In contrast, only 22.4% of the relevance scores are 0 for $\tau = 2$, which strikes a good balance between information linking and relevance differentiation.

RW Estimator Convergence. We evaluate the convergence rate of the proposed RW estimator on the connectivity score. Fig. 7 shows the average sampling error of $cdr_c(c, d)$ compared to the ground truth value. Solid and dotted lines represent RW *with* and *without* the guidance of k-hop reachability index respectively. With the k-hop index, our sampling approach can converge on all three datasets within 5% estimation error using 20 sampling iterations.

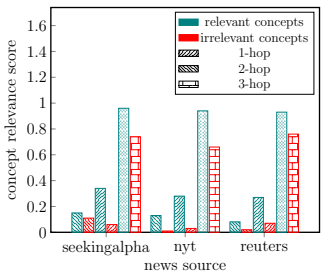

Fig. 6: Effectiveness study of context relevance

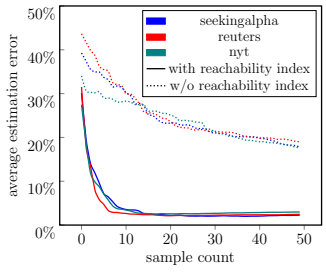

Fig. 7: Sampling error with increasing samples

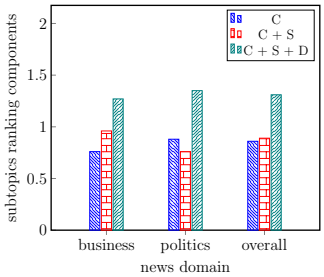

Fig. 8: Subtopics ablation study

5.5 Ablation Study on Drill-down Operations

Given a search query, NCExplorer automatically suggests related concepts for *drill-down* operations. NCExplorer ranks the concepts by

considering three key factors: *Coverage (C)*, *Specificity (S)* and *Diversity (D)*. To investigate the impact of each factor, we conduct user studies for the ablation analysis. We use the queries constructed from Sec. 5.2 and select top-ranked concepts when only considering: (1) C; (2) C+S; (3) C+S+D. We build an interactive survey interface (Fig. 9) that allows the participants to click on different concepts and view associated results before assigning a distinct rating 1-3 for each subtopic. We use the same platform to recruit participants and obtain 518 survey results in total. The results are displayed in Fig. 8. We can observe that the specificity contributes slight improvement to the overall rating while diversity plays a more significant role in rating improvement.

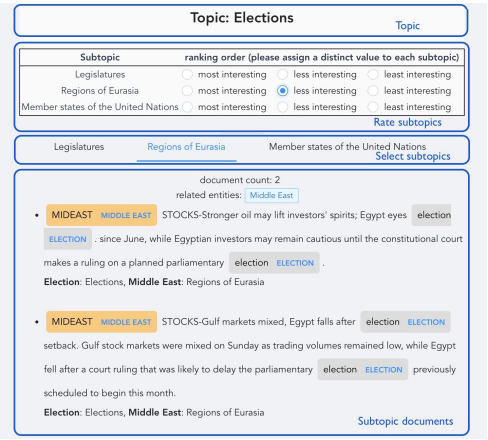

Fig. 9: Subtopic survey interface

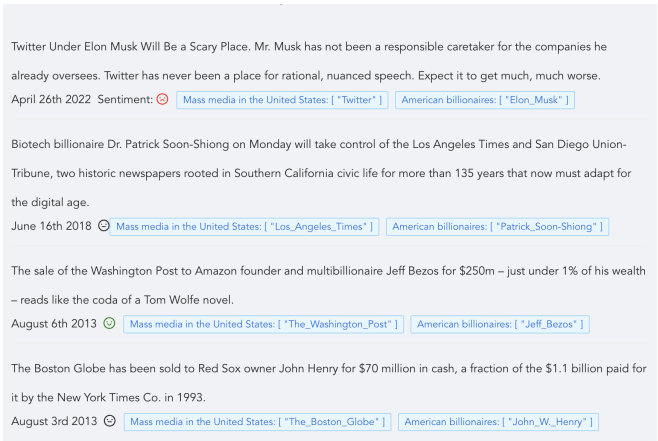

Fig. 10: Case study on Media Bias

5.6 Case Study on Media Bias

In this case study, NCExplorer's capabilities in identifying media bias are demonstrated. When Elon Musk announced his interest in acquiring Twitter, there were intense criticisms towards rich people controlling the media [8] With NCExplorer, users can roll-up "*Elon Musk*" and "*Twitter*" to a concept pattern query "*American billionaire*" and "*U.S. Mass Media Company*". Among all matched news of the query in the dataset, several similar acquisitions are discovered, such as Jeff Bezos acquiring the Washington Post, Patrick Soon-Shiong buying the Los Angeles Times, and Rupert Murdoch purchasing The Wall Street Journal. Furthermore, the sentiment score of each matched news article is evaluated using a pre-trained model [27]. Interestingly, Musk's acquisition is the only news with a negative sentiment. The rest have either neutral or positive sentiments (Fig 10). Such insights reveal a potential news bias that deserves further investigation by experts. This example showcases how NCExplorer can be a powerful tool for uncovering hidden patterns and biases in news coverage.

6 Conclusion

NCExplorer is an innovative tool designed to enrich the OLAP-like exploration experience of news articles by intelligently linking them to pertinent concepts within a knowledge graph. Its unique ranking system accounts for both concept relevance and article context, contributing to a more comprehensive understanding of the information.
To ensure computational efficiency, the tool employs a random walk estimator combined with a reachability index for context relevance ranking. Its effectiveness has been confirmed through crowd-sourced and GPT evaluations using real-world news data and a large-scale knowledge graph. The modular design of NCExplorer represents a significant contribution, as it enables seamless integration with various text systems such as search engines, news portals, and literature databases, ultimately enhancing information discovery and comprehension across multiple applications. A possible future direction is to extend the exploratory process beyond single news events to a sequence of evolving event series. This advancement could provide a more dynamic and extensive exploration experience, allowing users to analyze the development of news stories and better understand their implications over time.

References

1. Dbpedia snapshot 2021-06 release, https://www.dbpedia.org/blog/snapshot-2021-06-release/, accessed: 2023-02-22
2. The new york times, https://www.nytimes.com, accessed: 2023-02-22
3. Openai models gpt-4
4. Reuters, https://www.reuters.com, accessed: 2023-02-22
5. Seeking alpha, https://seekingalpha.com, accessed: 2023-02-22
6. Sentence transformers all-mpnet-base-v2, https://huggingface.co/sentence-transformers/all-mpnet-base-v2
7. Vector search engine, https://qdrant.tech/
8. Bensinger, G.: Twitter under elon musk will be a scary place, https://www.nytimes.com/2022/04/25/opinion/editorials/twitter-elon-musk.html, accessed: 2023-02-22
9. Cheng, J., Shang, Z., Cheng, H., Wang, H., Yu, J.X.: Efficient processing of k-hop reachability queries. The VLDB journal **23**(2), 227–252 (2014)
10. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding (2018)
11. Fernández, M., Cantador, I., López, V., Vallet, D., Castells, P., Motta, E.: Semantically enhanced information retrieval: An ontology-based approach. Journal of Web Semantics **9**(4), 434–452 (2011)
12. Flew, T., Spurgeon, C., Daniel, A., Swift, A.: The promise of computational journalism. Journalism practice **6**(2), 157–171 (2012)
13. Foundation, T.A.S.: Solr, https://solr.apache.org/, accessed: 2023-03-01

14. Gilardi, F., Alizadeh, M., Kubli, M.: Chatgpt outperforms crowd-workers for text-annotation tasks. arXiv preprint arXiv:2303.15056 (2023)
15. Honnibal, M., Montani, I.: spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. To appear **7**(1), 411–420 (2017)
16. Horvitz, D.G., Thompson, D.J.: A generalization of sampling without replacement from a finite universe. Journal of the American Statistical Association **47**(260), 663–685 (1952). https://doi.org/10.1080/01621459.1952.10483446
17. Ji, G., He, S., Xu, L., Liu, K., Zhao, J.: Knowledge graph embedding via dynamic mapping matrix. In: Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (volume 1: Long papers). pp. 687–696 (2015)
18. Kim, J., Nguyen, A.D., Lee, S.: Deep cnn-based blind image quality predictor. IEEE transactions on neural networks and learning systems **30**(1), 11–24 (2018)
19. Lehmann, e.a.: Dbpedia, a large-scale, multilingual knowledge base extracted from wikipedia. Semantic web (2015)
20. Li, F., Wu, B., Yi, K., Zhao, Z.: Wander join: Online aggregation via random walks. In: SIGMOD. pp. 615–629 (2016)
21. Liu, D., Bai, T., Lian, J., Zhao, X., Sun, G., Wen, J.R., Xie, X.: News graph: An enhanced knowledge graph for news recommendation. In: KaRS@ CIKM. pp. 1–7 (2019)
22. Liu, D., Lian, J., Liu, Z., Wang, X., Sun, G., Xie, X.: Reinforced anchor knowledge graph generation for news recommendation reasoning. In: KDD. pp. 1055–1065. ACM (2021)
23. Liu, D., Lian, J., Wang, S., Qiao, Y., Chen, J.H., Sun, G., Xie, X.: Kred: Knowledge-aware document representation for news recommendations. In: Proceedings of the 14th ACM Conference on Recommender Systems. pp. 200–209 (2020)
24. Mitra, G., Mitra, L.: The handbook of news analytics in finance, vol. 596. John Wiley & Sons (2011)
25. Niraj Chokshi, N.E.B.: Boeing and ford suspend operations in russia, https://www.nytimes.com/2022/03/01/business/boeing-ford-russia.html, accessed: 2023-02-22
26. Park, Y., Ko, S., Bhowmick, S.S., Kim, K., Hong, K., Han, W.S.: G-care: a framework for performance benchmarking of cardinality estimation techniques for subgraph matching. In: SIGMOD. pp. 1099–1114 (2020)
27. Pérez, J.M., Giudici, J.C., Luque, F.: pysentimiento: A python toolkit for sentiment analysis and socialnlp tasks (2021)
28. Qin, L., Peng, Y., Zhang, Y., Lin, X., Zhang, W., Zhou, J.: Towards bridging theory and practice: hop-constrained st simple path enumeration. VLDB Endowment (2019)
29. Reimers, N., Gurevych, I.: Sentence-bert: Sentence embeddings using siamese bert-networks. arXiv preprint arXiv:1908.10084 (2019)
30. Richardson, J., Sallam, R., Schlegel, K., Kronz, A., Sun, J.: Magic quadrant for analytics and business intelligence platforms. Gartner ID G00386610 (2020)

31. Robertson, S., Zaragoza, H., et al.: The probabilistic relevance framework: Bm25 and beyond. Foundations and Trends® in Information Retrieval **3**(4), 333–389 (2009)
32. Sheu, H.S., Li, S.: Context-aware graph embedding for session-based news recommendation. In: RecSys. pp. 657–662 (2020)
33. Sun, S., Chen, Y., He, B., Hooi, B.: Pathenum: Towards real-time hop-constrained st path enumeration. In: SIGMOD. pp. 1758–1770 (2021)
34. Tanon, T.P., Weikum, G., Suchanek, F.: Yago 4: A reason-able knowledge base. In: ESWC. pp. 583–596 (2020)
35. Thurman, N.: Computational journalism. Forthcoming, Thurman (2018)
36. Wang, H., Zhang, F., Xie, X., Guo, M.: Dkn: Deep knowledge-aware network for news recommendation. In: Proceedings of the 2018 world wide web conference. pp. 1835–1844 (2018)
37. Yang, Y., Li, Y., Tung, A.K.: Newslink: Empowering intuitive news search with knowledge graphs. In: ICDE. pp. 876–887 (2021)
38. Zhao, Z., Christensen, R., Li, F., Hu, X., Yi, K.: Random sampling over joins revisited. In: SIGMOD. pp. 1525–1539 (2018)