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

No Author Given

No Institute Given

Abstract. Effective news exploration is a common but hefty task for journalists and analysts alike. This paper proposes NCExplorer, the first of its kind framework that brings OLAP-like operations to news exploration. Specifically, the *roll-up* operation allows users to select preferred concepts for exploring similar contents in a broader context, while the *drill-down* operation provides exploitation capability via suggesting subtopics to users. The operations are enabled by leveraging both the fact and the ontology network from external knowledge graphs (KGs). We propose a novel ranking scheme to link news with KG concepts. We also develop an efficient unbiased sampler to scale the rank computation on large KGs due to the prohibitively expensive cost of graph path enumerations. Experimental results from crowd-sourced participants on Amazon Mechanic Turk confirm that NCExplorer is preferred for news exploration over state-of-the-art news search approaches on real-world news datasets over different topic domains.

1 Introduction

How to effectively explore a large corpus of news has always been critical to many analytical tasks, especially for professional users. For example, a journalist needs to gather relevant news reports to develop in-depth analysis in a featured article or an editorial [19,4]. Likewise, a business operator heavily relies on latest updates and historical reports to hedge against volatile markets [10,14].

For structured data, this analytical requirement has been addressed by the OLAP methods where a user can iteratively perform *roll-up* operations to gain a bird's-eye view or *drill-down* operations to dig deeper into the details. This interactive flow is equally useful when applied to unstructured news data. Imagining that analyst Alice conducts a market research originated from a news article on "*Tesla Motors*". It is important for Alice to explore news not only involving Tesla explicitly but also related content on Tesla's competitive landscape, e.g., updates on other electronic vehicle (EV) players like "*Waymo*" and "*NIO*", or even traditional car makers. Such a requirement resembles a *roll-up* operation from "*Tesla*" to car makers. Further, Alice wants to investigate different development status across geographic regions like "*Europe*", "*North America*" and "*Asia*". Hence, a *drill-down* operation is also desired to narrow down the news content and discover market insights. Many commercial search engines now offer navigation features by recommending *phrases that people also search for*, but

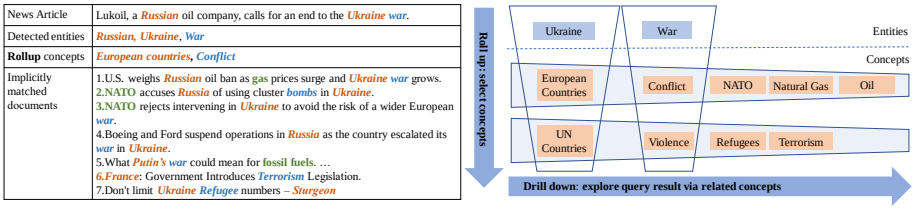

Fig. 1: Sample Exploration for NCExplorer. (best viewed in color)

the recommendations are mainly based on the popularity of searches and lack consistency in terms of systematic exploration for professionals.

To this end, we propose NCExplorer, a novel framework for systematic and effective news exploration. Given a news article, NCExplorer generates *semantic roll-up* and *drill-down* operations on the detected entities to support user exploration. In Fig. 1, a news article consists of two entities, "*Ukraine*" and "*War*". For each entity, NCExplorer generates related concepts to be rolled-up. For example, "*Ukraine*" can be generalized to "*European Countries*" or an even broader concept "*UN countries*". After users pick the rolled-up concepts, e.g., generalizing ("*Ukraine*", "*War*") to ("*European Countries*", "*Conflict*") respectively, NCExplorer returns relevant news and highlights how each result is linked to the rolled-up concepts. Further, NCExplorer finds novel subtopics from the relevant news for the drill-down operation. Among news matched to ("*European Countries*", "*Conflict*"), "*Natural Gas*", "*Oil*" and "*NATO*" are identified as subtopics, which reveals the conflicts reported in recent news have strong ties to the energy supply for users to investigate further.

To enable the *roll-up* and *drill-down* operations, NCExplorer leverages external knowledge graphs (KGs) [18,7]. These KGs contain not only millions of entities and relationships as a fact network but also rich ontology information. There are several technical challenges to be addressed in building NCExplorer.

The first challenge is how to effectively compute the relevance between a concept and a news document. Multiple concepts can be related to a news document. Taking the 4th document in Fig. 1 as an example, i.e., "*Boeing and Ford suspend operations in Russia.*" [1], "*Ford Motor Company*" matches several concepts according to the KG ontology such as "*Multinational Companies Headquartered in the United States*" and "*Motor Vehicle Manufacturers*". Nonetheless, "*Multinational Companies Headquartered in the United States*" is more relevant because it also generalizes another entity in the news: "*Boeing*". With this observation, we design a novel ranking function by considering 1) *ontology relevance* and 2) *context relevance*. Under the ontology relevance, a concept is considered relevant to a news document if the linked entity is important to the document. For context relevance, a concept's weight is determined from its relationship with remaining entities (a.k.a. context entities) in the news. However, the associations between a concept c and the context entities in the news are not captured by

[1] https://www.nytimes.com/2022/03/01/business/boeing-ford-russia.html

the KG ontology. Hence, we leverage the fact network in the KGs and evaluate the contextual relevance via the paths in the fact network connecting entities categorized by c with all context entities. Albeit its effectiveness, computing the contextual relevance is prohibitively expensive as it incurs path enumerations between many entity pairs. Thus, we scale NCExplorer by proposing an efficient random walk sampling algorithm with an unbiased estimation of the exact relevance score.

The second challenge is the ranking of concepts for the *drill-down* 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 devise 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 the concepts with higher specificity (fewer neighbors in KG) to penalize very 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 such that more distinct entities can be found in the *drill-down* result.

Our main technical contributions are summarized as below:

- We propose a novel framework NCExplorer, the first of its kind, to support news exploration with semantic *roll-up* and *drill-down* operations by leveraging the ontology and fact networks of KGs. (Sec. 3)
- We design effective ranking schemes to evaluate the relevance between concepts and news articles. We devise an efficient and unbiased sampling estimator for the relevance score computation. (Sec. 4)
- We evaluate NCExplorer on real-world news datasets with crowd-sourced participants from Amazon Mechanical Turk. Experiment results with more than 2500 responses with master qualification confirm that: the ranking schemes in NCExplorer are more effective than those of the state-of-the-art news search methods. Further, 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 semantic search that matches documents by meaning, beyond lexical similarities. While recent neural embedding approaches [2,13] are able to extract semantics, the embedding representation makes it difficult to interpret. Meanwhile, many efforts have been made to link text fragments with KGs [9,12] such that the semantics can be interpreted with KG entities. [5] transforms Wikipedia into an inverted index that maps each word into a list of weighted Wikipedia articles. The weighted $< word, article >$ vectors can represent text semantics. [6] derives semantics using links between Wikipedia articles. $< word, article >$ weight is calculated as the frequency of a word appearing within the size-k window of a Wikipedia article link. [3] uses the network

structure of KGs instead of the textual information. It takes a SPARQL query as input and uses the returned results as "*keywords*" for document matching. This approach enables a wide range of concept-based retrieval as long as the query can be written in the SPARQL syntax. This is similar to NCExplorer for queries with a single concept. However, for multi-concept queries, this approach requires all concepts to be connected in the KG. For example, the query "*legal disputes involve US banks that trade on NASDAQ*" yields empty result since the concepts "*NASDAQ*", "*Bank*", "*the U.S.*" are disconnected in the KG. Newslink [20] is the state-of-the-art news search approach that also exploits the network structure of KGs. It uses entities identified from a text fragment as the seed nodes. A graph expansion algorithm tries to connect the seed nodes into a single graph by adding hidden related nodes. The hidden nodes are then added to the text as auxiliary information for document matching. Newslink also requires the entities appearing in a text fragment to be connected in KGs to produce effective results. To our best knowledge, NCExplorer is the first work to enable systemic exploration with OLAP-ish operations. Although NCExplorer also takes advantage of the network structure of KG, our ranking scheme does not assume that the concepts or entities are connected.

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 [20], 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.

There are three major components in NCExplorer: *news indexing*, semantic *roll-up*, and *drill-down*:

News Indexing. Each news document $d \in \mathcal{D}$ is prepossessed by standard NLP techniques. 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 in 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/more entities with KG concepts to form

a concept pattern query Q. Given a query 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

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 to d by ontology relevance. Note that there could be more than one entity in d that matches c. We thus 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., $ME(c, d) = \{v |\ v \in d$ 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 [3,20], we do not assume that the entities in a document are connected. Finally, we normalize the connectivity score to $[0, 1)$ as: $cdr_c(c, d) = 1 - \frac{1}{1+conn(c,d)}$.

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 Fig. 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 scoring function, $sbr(\cdot, Q)$, over candidate subtopics and only suggest top-scored subtopics for Q. There are three key perspectives considered in $sbr(\cdot, Q)$ as below:

$$sbr(c, Q) = coverage(c, Q) \cdot specificity(c) \cdot diversity(c, Q) \tag{5}$$

- **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. Without 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., $diversity(c, Q) = \frac{|\bigcup_{d \in \mathcal{D}(Q)} ME(c,d)|}{|\mathcal{D}(Q \cup \{c\})|}$ where we normalize the score as the average number of distinct entities mapped to the concept c among all the documents relevant to $Q \cup \{c\}$.

4.3 Connectivity Score Estimation

Computing the connectivity score directly is prohibitively expensive for two reasons. First, consider the rightmost summation of Eqn. 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 only *polynomial delay* algorithms exist [17]. Second, there is a large number of 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 inverted index entry $< c, d, cdr(c, d) >$.

We thus devise a random walk estimator for 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:

$$r(u,v) = \mathcal{I}(u,v) \cdot |\Psi(c)| \cdot \beta^{l-1} \cdot \prod_{i=1}^{l-1} N(u_i) \tag{6}$$

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., $\widehat{conn}(c,d) = \frac{1}{\theta}\sum_{j=1}^{\theta} r_j$, which is an unbiased estimator. To enable faster convergence, we build a k-hop reachability index [1] on the KG instance space and only sample eligible neighbors that satisfy the hop constraint. Due to space limit, the detail algorithm is presented in our technical report [cite].

$r(\cdot,\cdot)$ 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.

5 Evaluation

In this section, we first present the experiment setup in Sec. 5.1, and then focus on answering 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 the proposed sampling method impact the accuracy? (Sec. 5.3)
3. How effective is each scoring component in the ranking model for the drill-down operation? (Sec. 5.4)
4. What kind of real analytical applications can NCExplorer enable (Sec. 5.5)?

5.1 Settings

Datasets. We use DBPedia[2] as our backend KG. Only the English version is used in our experiments, although NCExplorer is not language specific. The June 2021 snapshot we used 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*[3], *Seeking Alpha*[4] and *The New York Times*[5] to have a mixture of business and politics reports. The dataset is released together with the source code [6].

Compared Methods. We evaluate NCExplorer against the following methods with different news representation models.

[2] https://www.dbpedia.org/blog/snapshot-2021-06-release/
[3] https://www.reuters.com
[4] https://seekingalpha.com
[5] https://www.nytimes.com
[6] https://anonymous.4open.science/r/ncexplorer-1F32/

Table 1: Effectiveness of the compared methods. The best results are boldfaced and the second-best results are underlined.

Politics	Elections			Stock Market			International Relations		
	ncgd@1	ncgd@5	ncgd@10	ncgd@1	ncgd@5	ncgd@10	ncgd@1	ncgd@5	ncgd@10
Lucene	0.578	0.649	0.83	0.593	0.638	0.809	0.556	0.532	0.747
BERT	0.356	0.356	0.61	**0.746**	0.681	0.83	0.352	0.423	0.671
NewsLink	0.432	0.535	0.782	0.453	0.604	0.77	**0.935**	**0.898**	**0.953**
NewsLink-BERT	0.28	0.313	0.614	0.538	0.56	0.756	0.37	0.382	0.617
NCExplorer	**0.827**	**0.9**	**0.958**	0.709	**0.788**	**0.897**	0.889	0.848	0.943
Business	Corporate Finance			Lawsuits			Law Enforcement		
	ncgd@1	ncgd@5	ncgd@10	ncgd@1	ncgd@5	ncgd@10	ncgd@1	ncgd@5	ncgd@10
Lucene	0.539	0.718	0.854	0.188	0.208	0.624	0.522	0.583	0.792
BERT	0.73	0.752	0.873	0.125	0.128	0.557	0.739	0.519	0.743
NewsLink	0.488	0.653	0.815	0.292	0.286	0.69	0.397	0.625	0.796
NewsLink-BERT	0.9	0.802	0.911	0.229	0.304	0.705	0.397	0.506	0.733
NCExplorer	**0.938**	**0.926**	**0.971**	**0.958**	**0.98**	**0.983**	**0.886**	**0.893**	**0.96**

- Lucene [8] implements a typical bag-of-words keyword match model. We use BM25 [15] for the term weighting scheme with the default library settings. We also use Lucene's subproject Solr [16] as it supports the vector data type to implement the embedding models.
- Bert [2] is a popular neural text embedding model. We select the pre-trained model with 110M parameters. Each document is first transformed into a vector of 768 dimensions by Bert and then stored in Solr.
- Newslink [20] is the state-of-the-art implicit news retrieval 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-word 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 use Newslink to augment KG entities into the documents and the queries. The augmented documents and queries are then transformed into vectors by Bert.
- NCExplorer is the proposed approach in this paper. The parameters are set to $\tau = 2$ and $\beta = 0.5$ by defaults. The number of samples for connectivity score estimation is set to 50.

All compared methods except NCExplorer use the cosine similarity for news document relevance ranking.

Implementation. NCExplorer and Newslink are implemented in python 3.9. Bert and Lucene use the Solr implementation. 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

To validate the effectiveness of the concept document rank in NCExplorer, we recruit 100 crowd-sourced participants from Amazon Mechanical Turk plat-

form with Master Qualification[7] and ask them to choose a topic from *Politics* or *Business* domain. Each selected topic is combined with a country or an industry to simulate rolled-up queries such as *elections in African countries*, *lawsuits involve U.S. technology companies*. After selecting a topic, each participant is asked to rate 25 randomized documents retrieved by each of the five methods. To ensure a fair comparison, we hide the NCExplorer's result explanations . The relevance level is rated from 0 to 5 with 0.5 as incremental value. Among the total 100 responses, 25 participants selected business related topics and 75 selected politics related topics. In total 2500 ratings are obtained.

We use **NDCG@k** to evaluate the effectiveness of relevance ranking. The results in Table 1 confirm the superiority of the proposed concept document rank. NCExplorer achieves either the best or the second best results in all cases. NCExplorer presents bigger advantages on topics *Lawsuits* and *Law Enforcement* since news on these topics poses a higher barrier on users' domain knowledge for evaluation. The results have demonstrated that NCExplorer can effectively assist users in navigating news sources on professional topics.

5.3 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 first 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. 2 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. 3 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.

5.4 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

[7] https://blog.mturk.com/simplified-masters-qualifications-137d77647d1c

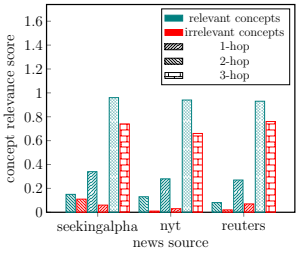

Fig. 2: Effectiveness study of context relevance

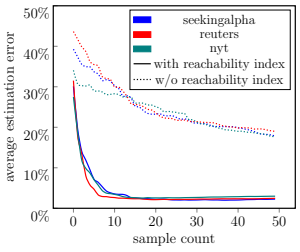

Fig. 3: Sampling error with increasing samples

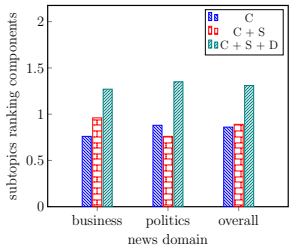

Fig. 4: Subtopics ablation study

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 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. 4. We can observe that specificity contributes slight improvement to the overall rating while diversity plays a more significant role in rating improvement.

5.5 Case Study on Media Bias

We conduct a case study to show the capabilities of NCExplorer on identifying media bias. When Elon Musk announced his interest in Twitter acquisition, there were intense criticisms towards rich people controlling the media[8]. By using 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 our dataset, we discover several similar acquisitions such as Jeff Bezo buys Washington Post, Patrick Soon-Shiong buys Log Angeles Times and Rupert Murdock buys The Wall Street Journal. Furthermore, we evaluate the sentiment score of each matched news via a pre-trained model [11] and find the Musk's acquisition is the only news with negative sentiment and the rest have either neural or positive sentiment.[9] [10] [11] Such insights reveal a potential news bias that deserves further investigations by experts.

6 Conclusion

We proposed NCExplorer to bring OLAP-like exploration to news analytics by linking news documents via KG concepts. We developed a novel concept doc-

[8] https://www.nytimes.com/2022/04/25/opinion/editorials/twitter-elon-musk.html

[9] https://twitter.com/BusinessInsider/status/364551288845254656

[10] https://www.latimes.com/business/hollywood/la-fi-ct-los-angeles-times-sold-20180207-story.html

[11] https://www.wsj.com/articles/SB118589043953483378

ument rank that considers both ontology and context relevance. To overcome expensive connectivity score computations in context relevance, we devised a random walk estimator paired with k-hop reachability index to accurately estimate the score with small samples. Extensive crowd-sourced evaluations have validated the effectiveness of NCExplorer on real-world news corpus and a large-scale KG. The modular design of NCExplorer can easily adapt to various text systems like search engines, news portals and literature databases.

References

1. 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)
2. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding (2018)
3. Fernández, M., Cantador, I., López, V., Vallet, D., Castells, P., Motta, E.: Semantically enhanced information retrieval: An ontology-based approach. J. Web Semant. **9**(4), 434–452 (2011)
4. Flew, T., Spurgeon, C., Daniel, A., Swift, A.: The promise of computational journalism. Journalism practice **6**(2), 157–171 (2012)
5. Gabrilovich, E., Markovitch, S., et al.: Computing semantic relatedness using wikipedia-based explicit semantic analysis. In: IJCAI. vol. 7, pp. 1606–1611 (2007)
6. Hassan, S., Mihalcea, R.: Semantic relatedness using salient semantic analysis. In: AAAI. vol. 25, pp. 884–889 (2011)
7. Lehmann, e.a.: Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. Semantic web (2015)
8. McCandless, M., Hatcher, E., Gospodnetić, O., Gospodnetić, O.: Lucene in action, vol. 2. Manning Greenwich (2010)
9. Mendes, P.N., Jakob, M., García-Silva, A., Bizer, C.: Dbpedia spotlight: shedding light on the web of documents (2011)
10. Mitra, G., Mitra, L.: The handbook of news analytics in finance, vol. 596. John Wiley & Sons (2011)
11. Pérez, J.M., Giudici, J.C., Luque, F.: pysentimiento: A python toolkit for sentiment analysis and socialnlp tasks (2021)
12. Rehurek, R., Sojka, P.: Software framework for topic modelling with large corpora. In: LREC workshop (2010)
13. Reimers, N., Gurevych, I.: Sentence-bert: Sentence embeddings using siamese bert-networks. arXiv preprint arXiv:1908.10084 (2019)
14. Richardson, J., Sallam, R., Schlegel, K., Kronz, A., Sun, J.: Magic quadrant for analytics and business intelligence platforms. Gartner ID G00386610 (2020)
15. Robertson, S., Zaragoza, H.: The probabilistic relevance framework: BM25 and beyond. Now Publishers Inc (2009)
16. Shahi, D.: Apache solr. Springer (2016)
17. Sun, S., Chen, Y., He, B., Hooi, B.: Pathenum: Towards real-time hop-constrained st path enumeration. In: SIGMOD. pp. 1758–1770 (2021)
18. Tanon, T.P., Weikum, G., Suchanek, F.: Yago 4: A reason-able knowledge base. In: ESWC. pp. 583–596 (2020)
19. Thurman, N.: Computational journalism. Forthcoming, Thurman (2018)
20. Yang, Y., Li, Y., Tung, A.K.: Newslink: Empowering intuitive news search with knowledge graphs. In: ICDE. pp. 876–887 (2021)