Diversified Query Generation Guided by Knowledge Graph

Xinyao Shen^{\bullet}, Jiangjie Chen^{\bullet}, Jiaze Chen^{\circ}, Chun Zeng^{\bullet}, Yanghua Xiao^{$\bullet \diamond \dagger$}

*Shanghai Key Laboratory of Data Science, School of Computer Science, Fudan University

[♡]ByteDance AI Lab

[◊]Fudan-Aishu Cognitive Intelligence Joint Research Center

{xinyaoshen19, jjchen19, zengc20, shawyh}@fudan.edu.cn, chenjiaze@bytedance.com

ABSTRACT

Relevant articles recommendation plays an important role in online news platforms. Directly displaying recalled articles by a search engine lacks a deep understanding of the article contents. Generating clickable queries, on the other hand, summarizes an article in various aspects, which can be henceforth utilized to better connect relevant articles. Most existing approaches for generating article queries, however, do not consider the *diversity* of queries or whether they are appealing enough, which are essential for boosting user experience and platform drainage. To this end, we propose a Knowledge-Enhanced Diversified QuerY Generator (KEDy), which leverages an external knowledge graph (KG) as guidance. We diversify the query generation with the information of semantic neighbors of the entities in articles. We further constrain the diversification process with entity popularity knowledge to build appealing queries that users may be more interested in. The information within KG is propagated towards more popular entities with popularity-guided graph attention. We collect a news-query dataset from the search logs of a real-world search engine. Extensive experiments demonstrate our proposed KEDy can generate more diversified and insightful related queries than several strong baselines. Our code is available at https://github.com/XinyaoShen/KEDY.

CCS CONCEPTS

- Computing methodologies \rightarrow Natural language generation.

KEYWORDS

Query generation; Information retrieval; Knowledge graph

ACM Reference Format:

Xinyao Shen, Jiangjie Chen, Jiaze Chen, Chun Zeng and Yanghua Xiao. 2022. Diversified Query Generation Guided by Knowledge Graph. In *Proceedings* of the Fifteenth ACM International Conference on Web Search and Data Mining (WSDM '22), February 21–25, 2022, Tempe, AZ, USA. ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3488560.3498431

[†]Corresponding author.

WSDM '22, February 21–25, 2022, Tempe, AZ, USA

© 2022 Association for Computing Machinery.

ACM ISBN 978-1-4503-9132-0/22/02...\$15.00

https://doi.org/10.1145/3488560.3498431

Table 1: An example (translated) of query generation task. Note that many entities (blue) mentioned in the article appear in the queries.

Title

| Appearance | comparison | photos | of Hollywood stars |
|------------|------------|--------|--------------------|
| | | | |

Content

In the past ten years, movie companies headed by Marvel and DC have almost maintained a tempo of 2-4 movies a year. They have also brought us such box office and good word-of-mouth double-harvest works as "Iron Man", "Avengers", "Aquaman" and "Spider-Man" further sweeping the American comics super hero craze to every corner of the world. In today's issue, I will bring you the appearance comparison photos of the actors starring in the American comics super hero movies so that you can understand the connotation of talent excellence. The appearance of Hollywood stars changes such as Jason Momoa who starred in "Aquaman". Jason's sturdy figure does not need to be said, and his performance in "Aquaman" has really shone the audience...

Query

1 INTRODUCTION

Diversified relevant articles recommendation is important for online news platforms to provide interesting information to users. Traditionally, users express their intents in a search engine by specifying queries expecting articles of interest to be retrieved, which is a classical *information retrieval* (IR) problem. In this paper, we study a reverse problem of IR, namely, *generating queries from an article* that lead to the click-through of an article.

Table 1 gives an example input article that contains a title and the corresponding article contents, as well as the possible generated queries. As seen in Table 1, search queries are more fine-grained than keywords and topics, providing a unique angle to reveal user interests; but more general than a headline or abstract of the document, since a query can cover a number of matched articles. In real applications, queries are usually placed at the bottom of an article for users to click on to browse more relevant articles. Thus, the recommendation of generated queries relieves users from the burden of article searching, which improves users' experience and stimulating users' activities on the platforms.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

 ⁽¹⁾ European and American star appearance
 (2) Hollywood star appearance ranking
 (3) Top 10 Marvel beauties
 (4) Spider-Man actor Tom Holland's new romance

Existing solutions mainly build queries for articles in terms of click-through data. If real users click on an article through a specific query, then this query is a good match for this article, because it successfully stimulates the interest of users. However, such naive methods rely heavily on user search behavior data, generating queries limited in diversity for the following reasons. First, generating effective queries for less popular articles is difficult, since they are rarely searched and do not have sufficient click-through logs. Second, an article is mainly recalled by popular queries. As a result the queries usually have rather limited diversity.

Further research employs generative models for producing more queries beyond user search data, posing problems in terms of generation accuracy and diversity. As articles are usually in the form of long text, traditional sequence-to-sequence (seq2seq) models [1, 23] have difficulty in capturing key information and being faithful to long articles. Recent studies [6, 17] alleviate this problem by transforming the input article into a graph structure, where each node represents key information of articles such as entities or sentences. However, the queries generated by these methods are usually too faithful to the input articles to encourage a wide range of interests of different users.

We argue that diverse but related queries can improve user experience. Such information is beyond the article but related to it so that users have a greater chance of being interested in them. Take the case in Table 1 for example. Queries like "*Spider-Man actor Tom Holland's new romance*" (query 4) might lead to the click-through of users who are fans of "*Tom Holland*", though "*Tom Holland*" is not mentioned in the article. However, there is little consideration of query diversity in search engines nowadays.

In order to generate more diversified queries, we argue that it is important to incorporate external knowledge graphs (KGs) for semantically relevant knowledge. However, a challenge arises that we have too many related entities in the semantic neighborhood to choose from in order to build diversified queries. As queries containing more popular knowledge are expected to attract more users, as a key perspective, we believe it is beneficial to utilize the *popularity* information for entities in a KG. The popularity information in a KG is usually represented by the click-through rate of an entity, which naturally reflects the popularity of an entity. For example, query 4 in Table 1 is appealing to users due to the popular entities "Spider-Man" and "Tom Holland". Recent studies leverage external KGs as extra semantic representations and additional input to the text generation models, which effectively improves the quality of generated text [5, 35]. However, in contrast, they treat knowledge in the KGs indiscriminately and rely on human annotation data to train knowledge selection.

In this paper, we propose KEDY, a Knowledge-Enhanced Diversified QuerY Generator. Specifically, we propose a novel graphical representation of the article called *entity interaction graph* to better take advantage of the structure of an article. To generate more diversified queries in terms of mentioned entities, we then incorporate multi-hop sub-graphs of a KG into query generation. Moreover, to further select popular information to be displayed in the queries, KEDY leverages *popularity-guided graph attention* on the relationships between entities. In this way, our model captures semantically relevant and appealing knowledge for diversified query generation. • We are the first to introduce knowledge graph into a generative model (KEDy) for diversified query generation.

- We guide the information propagation in the knowledge graph with popularity knowledge of entities for generating more appealing queries.
- We achieve state-of-the-art performance on a news-query dataset collected from a real-world platform in both automatic and human evaluation.¹

2 RELATED WORKS

Query Matching and Generation. As an early and popular attempt for connecting queries for given articles, query matching methods choose the best article for the query. Trivial methods compute cosine or TF-IDF similarity scores between query and article word embeddings. Many deep matching models have also been proposed [7, 8, 20, 24, 29], which are either representation-based that focus on semantic feature extraction, or interaction-based that emphasize pair-wise matching. Recent neural frameworks for text matching, despite their success in sentence matching, question answering, or query article matching, do not perform well at matching long articles. To alleviate this problem, Liu et al. [19] propose a topic interaction graph approach targeting long article matching, which motivates our design of article modeling.

For query generation, Wang et al. [32] propose a novel multi-task learning approach for title compression and query generation. Han et al. [6] propose a Graph-augmented Sequence to Attention (G-S2A) to make the most of article information for query generation. However, they do not consider the diversity and novelty of generated queries. We, in this work, incorporate an external knowledge graph to generate multiple more diversified and attractive queries.

Knowledge-injected Generation. Recently, pre-trained language models (PLMs), such as BERT [4], GPT-2 [26] and BART [13], further boost researches on natural language generation through large scale pre-training. Nevertheless, implicit knowledge in PLMs is not enough to help us generate diversified and informative text.

Incorporating explicit knowledge in Natural Language Generation (NLG) beyond input text is seen as a promising direction in both academia and industry [37]. Koncel et al. [11] introduce a graph transformer model that can leverage the relational structure of knowledge graphs. Yang et al. [36] investigate the potential of leveraging external knowledge bases (KBs) to further improve BERT for machine reading comprehension (MRC). Recently, there is a wide range of research work about question generation over knowledge graphs [2, 12, 27]. Xu et al. [35] use large-scale language models and add control to text generation by incorporating an external knowledge base. Li et al. [16] focus on inferring aspect information using KG data for diversified review generation. These mentioned knowledge-enhanced models including Knowledge-enhanced GPT-2 [5] and Facebook's RAG [14] do not control what knowledge to integrate or pay attention to the popularity information of entities. Different from previous research, KEDy uses a novel attention mechanism on all entities guided with popularity information.

¹ The dataset will be partially disclosed to the extent permitted.



Figure 1: An overview of KEDY Model. The whole model contains three main components: Graph Construction, Graph Representation Learning and Diversified Generation.

3 OUR PROPOSED APPROACH

The framework of our solution is illustrated in Fig. 1. It contains three main components: *Graph Construction, Graph Representation Learning* and *Diversified Generation.*

3.1 Graph Construction

In this section, we introduce how to construct the entity interaction graph from a news article and construct the knowledge sub-graphs.

3.1.1 Article Graph.

To fully understand the long news article, we employ a graph structure modeling news article. First, we do word segmentation and Named Entity Recognition (NER) to extract entities from the original article. In general, the recognized entities can hardly cover the complete information of the article. We further apply keywords extraction algorithms like TextRank [22]) to find additional keywords serving as complementary entities as well.

After we get the entities of the news article, we divide the article into sentences and number each sentence as $\{s_1, s_2, ..., s_n\}$. We assign sentence s_j to entity e_i if e_i appears in s_j . Each sentence may be assigned to multiple entities if it contains them. The sentences together with the entity e they belong to form the node n_e in the entity interaction graph. And we assign sentences that do not contain any entities to a special node *Empty*. What's more, there is also a special node called *Title*. The edge weight between entity nodes depends on the number of sentences shared between the two nodes.

3.1.2 Central Graph and Multi-hop Graph.

We also extract a central graph and a multi-hop graph from a large knowledge graph to guide query generation. We link the entities mentioned in the article to KG and refer to them as zero-hop entities, denoted by V^0 . We grow zero-hop entities V^0 with one-hop entities V^1 and two-hop entities V^2 . The central graph \mathbb{G}_C is defined as the sub-graph of the knowledge graph K consisting of the entities in $V^0 \cup V^1$ and relations between them. \mathbb{G}_C contains knowledge that is closely relevant to the article. Similarly, the multi-hop graph \mathbb{G}_M is the sub-graph induced by $V^1 \cup V^2$. Since there are many entities in the multi-hop graph and some of them are not necessarily relevant to the original article, a special attention mechanism will be employed in our solution (see Section 3.2.2).

3.2 Graph Representation Learning

In this section, we elaborate on how to encode the article graph, the central graph, and the multi-hop graph respectively.

3.2.1 Article Graph Encoding.

We use the self-attention mechanism to encode each node in the entity interaction graph. In particular, we label the text in each node as *T*. For each word in the text *T*, we use its word embedding w_i . And in order to highlight the special status of the entities in the long text *T*, we add the position embedding p_i of each word, and all the entities share the same position embedding p_0 . So the final embedding of each word is $\epsilon_i = w_i + p_i$. Then we use self-attention to model connections between words and make full use of contextual information of sentences. Then we get the hidden vector a_i of each word. The Scaled Dot-Product attention is as follows:

Attention(Q, K, V) =
$$\sigma(\frac{QK^T}{\sqrt{d_k}})V$$
 (1)

where **Q**, **K**, **V** are queries of dimension d_k , keys of dimension d_k and values of dimension d_v , and σ is the softmax function. Since the entity is the most important information in the node, we use

the hidden vector of the entity a_0 in the last layer as the vector that represents the whole node.

After we get the hidden vector of each node n_i in the article graph, we feed them to a graph encoder to make use of the graph structure of the entity interaction graph. We use spectral-based graph convolutional model (GCN) [10] because GCN can both model the information of each node and the graph structure. GCN iteratively convolves on the features of a node and its neighbors, which is defined by the following update rule:

$$\mathbf{H}^{(l+1)} = \sigma(\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{H}^{(l)}\mathbf{W}^{(l)})$$
(2)

where $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}_N$ is the adjacency matrix of the undirected graph with added self-connections, \mathbf{I}_N is the identity matrix. $\tilde{\mathbf{D}}_{ii} = \sum_j \tilde{\mathbf{A}}_{ij}$, $\mathbf{H}^{(l)} \in \mathbb{R}^{(N \times D)}$ is the matrix of activations in the *l*-th layer and $\mathbf{W}^{(l)}$ is a layer-specific trainable weight matrix.

Considering the strong relationship between title and queries in most cases, we use the hidden embedding of the title node as the initial state of the decoder to emphasize the importance of the title.

3.2.2 Knowledge Sub-graph Encoding.

The constructed knowledge sub-graphs including the central graph and the multi-hop graph provide explicit knowledge related to the article. We use a graph neural network to encode the central graph which is closely relevant to the article and propose a novel popularity-guided graph attention mechanism to guide the selection of knowledge entities in the multi-hop graph.

Central Graph Encoding. The central graph is encoded by a graph neural network GraphNet [30], which exhibits great effectiveness in encoding knowledge graphs. The *l*-th layer representation $\mathbf{g}_{e_i}^l$ of entity e_i is calculated by a single-layer feed-forward network (FFN):

$$\mathbf{g}_{e_{i}}^{l} = FFN(\mathbf{g}_{e_{i}}^{l-1} \circ \mathbf{h}^{l-1} \circ \sum_{r} \sum_{e_{j}} f_{r}^{e_{j} - > e_{i}}(\mathbf{g}_{e_{j}}^{l-1}))$$
(3)

where \circ is a concatenation operator and $\mathbf{g}_{e_i}^{l-1}$ is the (l-1)-th layer representation of entity e_i . \mathbf{h}^{l-1} is the (l-1)-th layer representation of the article, which is updated with the zero-hop entities V^0 :

$$\mathbf{h}^{l-1} = FFN(\sum_{e_i \in V^0} \mathbf{g}_{e_i}^{l-1}) \tag{4}$$

 $\mathbf{g}_{e_i}^0$ is initialized with the pre-trained entity embedding e_i . The article representation h^0 is initialized with the final hidden state of the title node in the article graph (see Section 3.2.1). $f_r^{e_j - > e_i}(\mathbf{g}_{e_j}^{l-1})$ aggregates the semantic information of each neighbor entity e_j with relation r.

Popularity-guided Graph Attention. We propose a popularityguided graph attention mechanism to encode the multi-hop graph so that more attention is paid to important entities since there are a lot of two-hop entities. The importance of entities contains two aspects: (1) the semantic relevance of the article. (2) the popularity information of entities which can improve the attraction of queries. Popularity information of entities is essential for query generation tasks since users intend to click on the queries containing more popular entities. For example, "Spider-Man actor Tom Holland's new romance" is a high-quality query since it contains popular entities "Spider-Man" and "Tom Holland" that most users are interested in. Therefore, we design this popularity-guided graph attention mechanism to satisfy these two requirements. Each entity has the property *views_times*. We rank the property *views_times* into *n* grades. The popularity score (**PopS**) is computed by the *views_times* property of the entities since highly viewed entities are quite likely to be popular and attractive. The attention η_r^{eq} leverages both relation embeddings **r** and the popularity score λ_{e_n} :

$$\eta_r^{e_q} = \sigma(\mathbf{P}^T \cdot \tanh(\mathbf{W}_p \cdot \mathbf{e}_p + \mathbf{W}_q \cdot \mathbf{e}_q))$$
(5)

$$\mathbf{P} = \mathbf{W}_r \cdot \lambda_{e_a} \mathbf{r} \tag{6}$$

$$\lambda_{e_q} = \frac{e_{vt} - s_k}{n \cdot \tau} + \frac{k}{n} \tag{7}$$

where **r** is the relation embedding between the entity $e_p \in V^1$ and its neighbor entity $e_q \in V^2$. \mathbf{W}_r , \mathbf{W}_p , \mathbf{W}_q are training parameters. \mathbf{e}_p and \mathbf{e}_q are embeddings for entity e_p and entity e_q . e_{vt} is the *views_times* of the entity. s_k is the number of the *k*-th grade. τ is the interval length between each of the two grades. We get an effective attention specially in terms of the relations and popularity information in multi-hop entities. Then we use this attention to aggregate triple (e_p, r, e_q) to get $\mathbf{p}_{\mathbf{e}_p}$:

$$\mathbf{p}_{\mathbf{e}_{p}} = \sum_{e_{q}} \eta_{r}^{e_{q}} \cdot [\mathbf{e}_{p} \circ \mathbf{e}_{q}]$$
(8)

where \circ is a concatenation operator. In the Popularity-guided Graph Attention module of Fig. 1, we can see our model KEDY pay more attention to popular two-hop neighbor entities (darker purple indicates higher attention score).

3.3 Diversified Generation

To consider both the article and knowledge information, the hidden representations of articles and knowledge computed by Section 3.2 are incorporated into the decoder to generate diversified queries.

3.3.1 Context Representation.

We use an attention-based LSTM decoder. On each time-step t, the decoder receives the previous decoder state s_{t-1} , context representation c_{t-1} and the word embedding of the previous word decoder emitted q_{t-1} :

$$\mathbf{s}_t = LSTM(\mathbf{s}_{t-1}, [\mathbf{c}_{t-1} \circ \mathbf{q}_{t-1}])$$
(9)

where \circ is a concatenation operator.

The context representation \mathbf{c}_{t-1} concatenates the article representation \mathbf{c}_{t-1}^A , the central graph representation \mathbf{c}_{t-1}^C and the multi-hop graph representation \mathbf{c}_{t-1}^M :

$$\mathbf{c}_{t-1} = FFN([\mathbf{c}_{t-1}^{A} \circ \mathbf{c}_{t-1}^{C} \circ \mathbf{c}_{t-1}^{M}]).$$
(10)

The context representations of the article, the central graph and the multi-hop graph read the hidden representations in Section 3.2 with a standard attention mechanism respectively:

$$\mathbf{c}_{t-1}^{A} = \sum_{i=1}^{n} \alpha_{t-1}^{i} \cdot \mathbf{h}_{i},$$

$$\mathbf{c}_{t-1}^{C} = \sum_{e_{i} \in \mathbb{G}_{C}} \beta_{t-1}^{e_{i}} \cdot \mathbf{g}_{e_{i}},$$

$$\mathbf{c}_{t-1}^{M} = \sum_{e_{p} \in \mathbb{G}_{M} \cap V^{1}} \gamma_{t-1}^{p} \cdot \mathbf{p}_{e_{p}}.$$
(11)

Table 2: Statistics of the dataset. Avg denotes the average token number of text. Ent. means Entertainment and Spt. means Sport.

| | Туре | Train | Dev | Test | Avg |
|------|---------|-----------|--------|--------|-------|
| Ent. | Article | 800,000 | 6,000 | 6,000 | 576.1 |
| | Query | 6,998,328 | 44,307 | 41,020 | 7.9 |
| Spt. | Article | 500,000 | 6,000 | 6,000 | 523.3 |
| | Query | 3,232,118 | 41,382 | 46,983 | 7.7 |

3.3.2 Diversified Token Generation.

The *t*-th time of the decoder hidden state \mathbf{s}_t calculated by Eq. 9 combines information from both the articles, the entities in the knowledge graph within two hops, and the attention upon them. The initial state of the decoder s_0 is the hidden embedding of the title node encoded in Section 3.2.1. The decoder leverages \mathbf{s}_t to generate the *t*-th token q_t to form the expected queries. Diversified queries contain more different tokens. Considering the tokens may be from different sources, we use a control gate μ^* to control the generation by choosing words from vocabulary ($\mu^* = 0$), central graph ($\mu^* = 1$, $V^0 \cup V^1$) and multi-hop graph ($\mu^* = 2$, V^2).

$$\mu^* = \arg\max_{\mu \in \{0,1,2\}} FFN_{\mu}(\mathbf{s}_t).$$
(12)

Then we generate tokens according to different control gate μ^* . The generation probabilities of words w, entity e_i in \mathbb{G}_C , multi-hop entity e_q in multi-hop graph are computed by Eq. 13:

$$q_t = \begin{cases} \sigma(\mathbf{s}_t \cdot \mathbf{w}), & \mu^* = 0\\ \sigma(\mathbf{s}_t \cdot \mathbf{g}_{e_i}), & \mu^* = 1\\ \sigma(\mathbf{s}_t \cdot \mathbf{e}_q), & \mu^* = 2 \end{cases}$$
(13)

where **w** is the word embedding of word w, g_{e_i} is the central graph representation of $e_i \in \mathbb{G}_C$ and \mathbf{e}_q is the entity embedding of the two-hop neighbor entity e_q .

4 EXPERIMENTS

In this section, we present the dataset and experimental setup. Then we compare our model with several strong baselines on a real-world Chinese news-query dataset.

4.1 Dataset

Our ideal dataset contains articles and matching queries. Since there is no appropriate publicly available dataset, we collect a Chinese news-query dataset from query-article pairs in the click-through logs of a popular real-world search engine. The article a user clicks and spends some time reading can be viewed as a positive match to the query, that is, a query-article pair. Considering the size of the whole dataset is very large, we use specific domains (entertainment and sport) of this dataset. We keep queries whose lengths are within the range of 5 to 10 in tokens. Train, development, and test sets are randomly divided and split as seen in Table 2. For the external knowledge base, we use CN-DBpedia [33], a large-scale Chinese general domain structured encyclopedia knowledge graph consisting of 67 million triples.

4.2 Experimental Setup

4.2.1 Implementation Details.

We take the top 100,000 most frequent words as vocabulary from the articles and queries. Glove embedding and TransE embedding are used to initialize the representations of the words and entities in KG. We use the embedding size of 128 and the batch size of 32. Word embeddings are shared between encoder and decoder. The number of multi-head attention heads is set to 8 and the hidden size is set to 128. The number of GNN layers is 2. We use Adam optimizer [9] with a learning rate of 0.001 to train the parameters and train for 20 epochs on a Tesla-V100 GPU.

4.2.2 Metric.

We adopt both automatic and human metrics for evaluation. We adopt **BLEU** [25] and **ROUGE** [18], which are widely used in summarization and generation tasks. Considering the limitations of these two metrics in evaluating the quality of generation, we use more metrics for diversity evaluation. We define the average BLEU score to be the **Self-BLEU** of results regarding one generated query as the hypothesis and the others as references [40]. We also measure the diversity by the ratio of distinct uni-grams (**Dist-1**) and bi-grams (**Dist-2**) [15] in all generated queries. As the ratio itself can not reflect the frequency distribution of n-grams, we further calculate the entropy-based metric [38].

For human evaluation, we ask 20 raters to score on 500 generation results, and each result will be evaluated by 4 raters. The evaluation is conducted towards five different aspects: **Correlation**, **Diversity, Informativeness, Fluency and Novelty**.

4.2.3 Baselines.

We compare KEDY with a variety of baselines, ranging from traditional ranking-based methods to state-of-the-art pre-trained language models. The baselines include:

- **TextRank** [22] is a graph-based ranking model for text processing that can construct a word network by looking at which words follow one another to find the most relevant keywords in text.
- **PtrGen** [28] is an attention-based Seq2seq model which can copy words from the source text via pointing and use a coverage mechanism to discourage repetition.
- **Transformer** [31] is a very effective generative model proposed by [31] based solely on attention mechanisms, dispensing with recurrence and convolutions entirely.
- CVAE [39] is a conditional variational auto-encoder learning framework to maximize the data likelihood augmented with the KL-annealing technique[3] and a BOW loss [39].
- **DP-GAN** [34] assigns low reward for repeatedly generated text and high reward for diversified and fluent text.
- **BART** [13] is a denoising autoencoder for pre-training sequenceto-sequence models, which is trained by corrupting text with an arbitrary noising function and learning a model to reconstruct the original text.
- M-CNTRL [35] incorporates an external knowledge base to language models for story generation.
- **Graph2Seq** [17] is a graph-to-sequence model that models the input news as a topic interaction graph, which can better understand the internal structure of the whole article.

| Model | | Ente | rtainm | ent | | | | Sport | | |
|---------------------|---------------|------------|---------------|------|------|---------------|---------------|---------------|------|------|
| mouel | B-1 | B-2 | B-4 | R-1 | R-L | B-1 | B-2 | B-4 | R-1 | R-L |
| TextRank [22] | 22.6 | 11.6 | 0.9 | 26.3 | 22.5 | 22.5 | 11.4 | 0.8 | 26.1 | 22.4 |
| PtrGen [21] | 49.5 | 38.4 | 18.1 | 42.5 | 41.3 | 49.4 | 38.3 | 18.1 | 42.3 | 41.2 |
| Transformer [28] | 50.6 | 39.7 | 19.0 | 44.3 | 42.9 | 50.7 | 39.7 | 19.1 | 44.4 | 42.9 |
| Transformer+KG [28] | 50.9 | 39.9 | 19.2 | 44.3 | 43.0 | 50.8 | 39.7 | 19.1 | 44.6 | 43.0 |
| CVAE [39] | 50.7 | 39.7 | 19.1 | 44.1 | 42.9 | 50.7 | 39.9 | 19.1 | 43.8 | 42.6 |
| DP-GAN [34] | 51.0 | 39.9 | 19.0 | 44.2 | 42.9 | 50.9 | 39.8 | 18.8 | 44.2 | 42.8 |
| BART [13] | 51.7 | 40.6 | 20.8 | 46.5 | 44.2 | 51.7 | 40.6 | 20.7 | 46.5 | 44.1 |
| BART+KG [13] | 52.2 | 40.9 | 21.0 | 46.8 | 44.5 | 52.1 | 41.0 | 20.9 | 46.8 | 44.7 |
| M-CNTRL [35] | 52.7 | 41.2 | 20.9 | 47.1 | 44.8 | 52.9 | 41.3 | 21.2 | 47.4 | 45.1 |
| Graph2Seq [17] | 52.8 | 41.2 | 20.9 | 47.2 | 45.7 | 52.7 | 41.1 | 20.9 | 47.0 | 45.6 |
| G-S2A [6] | 53.1 | 41.3 | 20.5 | 47.5 | 46.1 | 53.0 | 41.2 | 20.5 | 47.3 | 46.1 |
| G-S2A+KG [6] | 53.8 | 41.6 | 20.8 | 47.8 | 46.2 | 53.6 | 41.5 | 20.8 | 47.7 | 46.3 |
| KEDy (Ours) | 56.9 * | 44.7^{*} | 23.9 * | 50.2 | 48.6 | 56.6 * | 44.6 * | 23.6 * | 50.5 | 48.5 |

Table 3: Automatic evaluation results from different models. B and R are short for BLEU and ROUGE. Significant improvements over the best baseline are marked with * (Wilcoxon signed-rank test, p < 0.01).

Table 4: Human evaluation results. Our model achieves the best score on almost all the aspects. Note that results of different models are similar on Correlation metric, and knowledge is helpful to improve Informativeness, Diversity and Novelty.

| Model | Entertainment | | | | | Sport | | | | | | |
|---------------------|---------------|------|------|------|------|-------|------|------|------|------|------|------|
| initial i | Cor | Div | Info | Flu | Nov | Avg | Cor | Div | Info | Flu | Nov | Avg |
| PtrGen [31] | 4.76 | 2.63 | 3.82 | 4.16 | 3.65 | 3.80 | 4.75 | 2.64 | 3.77 | 4.18 | 3.68 | 3.80 |
| Transformer+KG [28] | 4.83 | 2.65 | 3.87 | 4.04 | 3.71 | 3.83 | 4.84 | 2.65 | 3.95 | 4.01 | 3.64 | 3.82 |
| CVAE [39] | 4.75 | 3.02 | 3.95 | 4.08 | 3.73 | 3.91 | 4.77 | 3.03 | 4.01 | 4.03 | 3.81 | 3.93 |
| DP-GAN [34] | 4.76 | 3.01 | 3.92 | 4.11 | 3.71 | 3.90 | 4.75 | 3.05 | 4.04 | 4.05 | 3.79 | 3.94 |
| BART+KG [13] | 4.81 | 3.15 | 4.15 | 4.28 | 3.80 | 4.04 | 4.82 | 3.18 | 4.14 | 4.30 | 3.85 | 4.06 |
| M-CNTRL [35] | 4.80 | 3.24 | 4.13 | 4.52 | 3.81 | 4.10 | 4.82 | 3.25 | 4.12 | 4.55 | 3.86 | 4.12 |
| Graph2Seq [17] | 4.81 | 3.01 | 4.12 | 4.60 | 3.80 | 4.07 | 4.82 | 3.03 | 4.11 | 4.58 | 3.82 | 4.07 |
| G-S2A+KG [6] | 4.81 | 3.10 | 4.20 | 4.59 | 3.82 | 4.11 | 4.80 | 3.12 | 4.21 | 4.60 | 3.82 | 4.11 |
| KEDY (Ours) | 4.82 | 4.03 | 4.31 | 4.65 | 4.08 | 4.35 | 4.84 | 4.05 | 4.33 | 4.62 | 4.08 | 4.36 |

• **G-S2A** [6] is a novel generative model called the Graphaugmented Sequence to Attention network for search query generation. It contains a sentence-level GCN [10] and a keyword-level GCN [10] as well as a hierarchical recurrent neural network (RNN) to encode the article.

We categorize these baselines as follows: 1) *extractive methods* (TextRank); 2) *sequence-to-sequence generation* (PtrGen and Transformer); 3) *diversified generation* (CVAE and DP-GAN); 4) *generation based on pre-trained language models* (BART and M-CNTRL); 5) *generation based on graph neural network* (Graph2Seq and G-S2A, where G-S2A is the state-of-the-art method for this task). Additionally, to verify the effectiveness of incorporating KG in our model, we also integrate KG as additional inputs into Transformer, BART and G-S2A for fair comparison (i.e., Transformer+KG, BART+KG, G-S2A+KG). Here, we simply append the knowledge as the text form to the original input.

4.3 Results

We compare our model with several baselines. As is shown in Table 3, the graph structure can help us to notice many different aspects of the article and efficient use of knowledge can introduce more related information for more diversified queries.

KEDY outperforms all generative models on all metrics. We also note that keywords extractor TextRank [22] works far worse than other models, indicating the difference between query generation and keywords extraction. Graph2Seq [10] achieves better performance than the Seq2seq model, demonstrating the benefits of modeling long articles using graph structures. KEDv increases 6.0 BLEU-1 scores compared to Transformer+KG [31] and 4.7 BLEU-1 scores compared to BART+KG [13]. Note that Transformer+KG and BART+KG improve just a little compared to Transformer and BART, which proves that simply adding knowledge text to the input is invalid and the effectiveness of knowledge incorporation in our model. G-S2A [6] is the best model for query generation we know. Our model KEDv increases 3.1 BLEU-1, 3.1 BLEU-2 scores to G-S2A+KG [6].

| Table 5: Manual pair-wise comparisons between different |
|---|
| models, where KEDY clearly wins compared with all base- |
| lines. |

| Model | Win | Lose | Tie |
|-------------------------|-------|-------|-------|
| KEDy vs. Transformer+KG | 85.2% | 3.6% | 11.2% |
| KeDy vs. M-CNTRL | 75.6% | 9.0% | 15.4% |
| KEDY vs. G-S2A+KG | 71.2% | 11.4% | 17.4% |

Table 6: Automatic diversity evaluation results.

| Model | Self-BLEU-2 | Dist-1 | Dist-2 | Ent-2 |
|----------------|-------------|--------|--------|-------|
| Transformer+KG | 35.8 | 0.027 | 0.125 | 6.26 |
| M-CNTRL | 28.2 | 0.056 | 0.312 | 7.52 |
| G-S2A+KG | 27.5 | 0.067 | 0.321 | 7.23 |
| KeDy | 21.7 | 0.186 | 0.521 | 8.68 |

Human evaluation results are illustrated in Table 4. Generally, our model KEDy beats almost all the baselines for all metrics. In particular, we find that the performance gaps between our model and baselines are more obvious in Diversity, Informativeness and Novelty which our model focus on and these features are important for queries. Note that these improvements are difficult to achieve through the internal mechanism of models, the introduction of knowledge can significantly improve these metrics. Comparing KEDy with G-S2A+KG, the *p*-value of Wilcoxon signed-rank testing at 95% confidence level is 2.7e - 3, which means the improvements achieved by our approach are statistically significant. Furthermore, we asked people to choose the better one between 500 queries generated by different models on the entertainment dataset to simulate the user's behaviors when browsing the platform. From Table 5, we can see that our model wins in most cases, which means our model KEDy can generate more diversified and attractive queries to users.

4.4 Diversity Evaluation

Diversity is a major concern for query generation. Therefore, we conduct the diversity evaluation on the entertainment dataset for this task. We use Self-BLEU [40] to evaluate the diversity of queries. We also use Dist-1, Dist-2 and the entropy of bi-grams (Ent-2) to evaluate the results. As Table 6 illustrates, Transformer+KG performs worst as expected. M-CNTRL improves diversity through better-integrating knowledge and G-S2A+KG achieves this effect through modeling the news articles into a graph structure. Our model KEDY combines these two strengths to achieve the best results on diversity evaluation. Although our model also has improvements on BLEU and ROUGE, the improvement in diversity evaluation is more significant, showing the strong advantages of our model in diversified query generation.

Correlation between Popularity and Query Clicks. To verify the advantages of popularity knowledge incorporation, we statistically analyze the correlation between popularity knowledge and query clicks. We experiment on both real user data (10 million)



Figure 2: Visualization of the correlation between popularity and user-clicks of queries. As seen in the figures, they are positively correlated in general, both in real and generated data.



Figure 3: Statistics on popularity score and uniqueness of queries generated by different models, where KEDY generates more attractive and informative queries.

in the search engine and the generated data of our model. Specifically, we sample 100 queries at 25 intervals between 25 and 200 clicks. The popularity score (**PopS**) of each query, as calculated in Equation 7, represents the view times of the entities within a query. As shown in Fig. 2, we observe that queries that contain more popular entities would have higher click times on both real data and generated data. Therefore, these results further promote us to incorporate popularity entities for query generation.

Effectiveness of Popularity Knowledge Incorporation. The uniqueness of the query generation task determines its strong demand for knowledge. To evaluate the effectiveness of popularity knowledge incorporation, we compare the average popularity scores (**PopS**) of generated queries by different models. From Fig. 3a, we can see that our model KEDY achieves higher popularity scores which means it can generate queries containing more popular entities. Furthermore, we also count the number of "unique words" in the test outputs on both the entertainment and the sport dataset. "Unique words" are those not in the pre-defined stop word list. This can be seen as an indicator for diversity and informativeness of queries since the more unique words contained the more information expressed. Fig. 3b illustrates that our model generates queries containing more "unique words" which means they are more diversified and informative.

| Ablation | B-1 | PopS | Self-BLEU-2 |
|-----------------------|------|------|-------------|
| KeDy | 56.9 | 0.73 | 21.7 |
| w/o Popularity-guided | 53.9 | 0.57 | 21.5 |
| w/o Multi-hop | 54.2 | 0.52 | 24.2 |
| w/o KG | 52.9 | 0.31 | 27.8 |
| w/o Graph | 50.3 | 0.28 | 35.1 |

Table 7: Ablation of KEDy.

4.5 Ablation Study

To evaluate the advantages of the graph representations and knowledge incorporation, we further evaluate KEDY through ablation study on the entertainment dataset and report the results in Table 7. We prepare four variants for comparison: w/o **Popularityguided**: The variant removes the popularity-guided graph attention mechanism. w/o **Multi-hop**: The variant removes the multi-hop graph and only uses one-hop entities. w/o **KG**: The variant removes the incorporation of knowledge. w/o **Graph**: The variant removes knowledge sub-graphs and does not construct the article as a graph.

We observe that introducing multi-hop entities improves query diversity significantly, and popularity-guided graph attention increases BLEU-1 and the popularity score with almost the same Self-BLEU-2. We can see that removing the incorporation of knowledge decreases all the metrics which proves the necessity of knowledge in query quality and diversity. Besides, the variant removing the article graph decreases BLEU-1 which confirms our point that graphical representations for long articles are of great importance for query generation tasks since Seq2seq cannot make full use of the information of the long articles. In particular, KEDy decreases 13.4 Self-BLEU-2 compared to the basic model w/o Graph, which is a huge improvement in diversity.

4.6 Case Study

Table 8 shows an example of queries generated by different models. From specific examples, we can see queries generated by our proposed model are more informative, diversified and attractive.

We find that queries generated from Transformer+KG mostly express the same meanings, e.g. "Marvel and DC" and "Hollywood movies" have appeared many times in these models. BART+KG can generate more diversified queries since pre-trained models can capture richer textual information. M-CNTRL generates more various queries by effectively introducing knowledge, e.g. "Iron Man actor Robert Downey". G-S2A+KG pays attention to more diversified information since the topic interaction graph can focus on different aspects of the articles. By contrast, KEDy generates the most diversified and attractive queries focusing on different aspects after the effective incorporation of knowledge. "Analysis of Thanos Infinity Gauntlet" is generated because we pay more attention to the two-hop entity "Thanos Infinity Gauntlet" since it has a high popularity score. Furthermore, in addition to introducing new entities, knowledge can also help us strengthen the relationships between existing entities to generate more user-friendly queries. For example, since we strengthen the relationship between "Hollywood" and "Spider-Man 3", "Hollywood movie Spider-Man 3" is generated.

Table 8: Case Study. These are queries generated by different models (Translated). Each model generates 5 queries. Some special words are marked blue. The article is about "Appearance comparison photos of Hollywood stars" in Table 1.

| Model | Query | | | | | |
|----------------|--|--|--|--|--|--|
| | Superhero | | | | | |
| | Hollywood actress group photo | | | | | |
| Transformer+KG | Hollywood movies | | | | | |
| | Marvel and DC | | | | | |
| | Hollywood movie box office | | | | | |
| | Marvel actor photos | | | | | |
| | Hollywood actress pictures | | | | | |
| BART+KG | Iron Man in-depth analysis | | | | | |
| | Marvel's new films change schedule | | | | | |
| | Spider-Man 3 poster | | | | | |
| | Hollywood female star | | | | | |
| | Is Iron Man dead at last | | | | | |
| M-CNTRL | Hollywood movies box office | | | | | |
| | Iron Man actor Robert Downey | | | | | |
| | Hollywood star appearance ranking | | | | | |
| | Spider-Man protagonist | | | | | |
| | The difference between Marvel and DC | | | | | |
| G-S2A+KG | Iron Man's story | | | | | |
| | Hollywood movies | | | | | |
| | Hollywood star group photo | | | | | |
| | Hollywood movie Spider-Man 3 | | | | | |
| | Spider-Man actor Tom Holland's new romance | | | | | |
| KeDy | Top 10 Marvel beauties | | | | | |
| | Analysis of Thanos Infinity Gauntlet | | | | | |
| | Hollywood's most handsome male star | | | | | |

5 CONCLUSION

In this paper, we propose KEDy, a knowledge-enhanced method for diversified article query generation. To our best knowledge, this is the first approach that introduces knowledge graphs into this task for query diversity. Specifically, KEDy considers the multi-hop entities of each entity mentioned in articles and uses popularity-guided graph attention for selecting diversified but popular entities in the knowledge graph. Experiments on both automatic and human evaluation demonstrate that our approach can generate various and informative queries outperforming the state-of-the-art methods on this task. Detailed analysis shows that the incorporation of knowledge graphs greatly increases the diversity of generated queries. Moreover, we have shown the positive correlation between popularity and user-clicks of queries. Experiments also demonstrate the effectiveness of the guidance of popularity knowledge for generating more popular queries. Future work should explore how to conduct personalized generation considering user preferences and how to better exploit semantic features of high-click queries.

ACKNOWLEDGEMENTS

We thank the anonymous reviewers for their valuable comments and suggestions. This work is supported by National Key Research and Development Project (No.2020AAA0109302), Shanghai Science and Technology Innovation Action Plan (No.19511120400) and Shanghai Municipal Science and Technology Major Project (No.2021SHZDZX0103).

REFERENCES

- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473, 2014.
- [2] Sheng Bi, Xiya Cheng, Yuan-Fang Li, Yongzhen Wang, and Guilin Qi. Knowledgeenriched, type-constrained and grammar-guided question generation over knowledge bases. In Proceedings of the 28th International Conference on Computational Linguistics, pages 2776–2786, 2020.
- [3] Samuel Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. Generating sentences from a continuous space. In Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning, pages 10–21, 2016.
- [4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.
- [5] Jian Guan, Fei Huang, Zhihao Zhao, Xiaoyan Zhu, and Minlie Huang. A knowledge-enhanced pretraining model for commonsense story generation. *Transactions of the Association for Computational Linguistics*, 8:93–108, 2020.
- [6] Fred X Han, Di Niu, Kunfeng Lai, Weidong Guo, Yancheng He, and Yu Xu. Inferring search queries from web documents via a graph-augmented sequence to attention network. In *The World Wide Web Conference*, pages 2792–2798, 2019.
- [7] Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen. Convolutional neural network architectures for matching natural language sentences. In Advances in neural information processing systems, pages 2042–2050, 2014.
- [8] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. Learning deep structured semantic models for web search using clickthrough data. In Proceedings of the 22nd ACM international conference on Information & Knowledge Management, pages 2333–2338, 2013.
- [9] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.
- [10] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907, 2016.
- [11] Rik Koncel-Kedziorski, Dhanush Bekal, Yi Luan, Mirella Lapata, and Hannaneh Hajishirzi. Text generation from knowledge graphs with graph transformers. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 2284–2293, 2019.
- [12] Vishwajeet Kumar, Yuncheng Hua, Ganesh Ramakrishnan, Guilin Qi, Lianli Gao, and Yuan-Fang Li. Difficulty-controllable multi-hop question generation from knowledge graphs. In *International Semantic Web Conference*, pages 382–398. Springer, 2019.
- [13] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. arXiv preprint arXiv:1910.13461, 2019.
- [14] Patrick Lewis, Ethan Perez, Aleksandara Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. arXiv preprint arXiv:2005.11401, 2020.
- [15] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. A diversitypromoting objective function for neural conversation models. In *HLT-NAACL*, 2016.
- [16] Junyi Li, Siqing Li, Wayne Xin Zhao, Gaole He, Zhicheng Wei, Nicholas Jing Yuan, and Ji-Rong Wen. Knowledge-enhanced personalized review generation with capsule graph neural network. In Proceedings of the 29th ACM International Conference on Information & Knowledge Management, pages 735–744, 2020.
- [17] Wei Li, Jingjing Xu, Yancheng He, ShengLi Yan, Yunfang Wu, and Xu Sun. Coherent comments generation for chinese articles with a graph-to-sequence model. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 4843–4852, 2019.
- [18] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In Text summarization branches out, pages 74–81, 2004.
- [19] Bang Liu, Ting Zhang, Di Niu, Jinghong Lin, Kunfeng Lai, and Yu Xu. Matching long text documents via graph convolutional networks. arXiv preprint arXiv:1802.07459, pages 2793–2799, 2018.
- [20] Zhengdong Lu and Hang Li. A deep architecture for matching short texts. Advances in neural information processing systems, 26:1367–1375, 2013.

- [21] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. arXiv preprint arXiv:1508.04025, 2015.
- [22] Rada Mihalcea and Paul Tarau. Textrank: Bringing order into text. In Proceedings of the 2004 conference on empirical methods in natural language processing, pages 404–411, 2004.
- [23] Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gulçehre, and Bing Xiang. Abstractive text summarization using sequence-to-sequence rnns and beyond. In Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning, pages 280–290, 2016.
- [24] Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, Shengxian Wan, and Xueqi Cheng. Text matching as image recognition. In AAAI, volume 16, pages 2793–2799, 2016.
- [25] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In Proceedings of the 40th annual meeting of the Association for Computational Linguistics, pages 311–318, 2002.
- [26] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.
- [27] Apoorv Saxena, Aditay Tripathi, and Partha Talukdar. Improving multi-hop question answering over knowledge graphs using knowledge base embeddings. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 4498–4507, 2020.
- [28] Abigail See, Peter J Liu, and Christopher D Manning. Get to the point: Summarization with pointer-generator networks. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1073–1083, 2017.
- [29] Yelong Shen, Xiaodong He, Jianfeng Gao, Li Deng, and Grégoire Mesnil. Learning semantic representations using convolutional neural networks for web search. In Proceedings of the 23rd international conference on world wide web, pages 373–374, 2014.
- [30] Haitian Sun, Bhuwan Dhingra, Manzil Zaheer, Kathryn Mazaitis, Ruslan Salakhutdinov, and William Cohen. Open domain question answering using early fusion of knowledge bases and text. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 4231–4242, 2018.
- [31] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Advances in neural information processing systems, pages 5998–6008, 2017.
- [32] Jingang Wang, Junfeng Tian, Long Qiu, Sheng Li, Jun Lang, Luo Si, and Man Lan. A multi-task learning approach for improving product title compression with user search log data. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.
- [33] Bo Xu, Yong Xu, Jiaqing Liang, Chenhao Xie, Bin Liang, Wanyun Cui, and Yanghua Xiao. Cn-dbpedia: A never-ending chinese knowledge extraction system. In International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, pages 428–438. Springer, 2017.
- [34] Jingjing Xu, Xuancheng Ren, Junyang Lin, and Xu Sun. Diversity-promoting gan: A cross-entropy based generative adversarial network for diversified text generation. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 3940–3949, 2018.
- [35] Peng Xu, Mostofa Patwary, Mohammad Shoeybi, Raul Puri, Pascale Fung, Animashree Anandkumar, and Bryan Catanzaro. Controllable story generation with external knowledge using large-scale language models. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 2831–2845, 2020.
- [36] An Yang, Quan Wang, Jing Liu, Kai Liu, Yajuan Lyu, Hua Wu, Qiaoqiao She, and Sujian Li. Enhancing pre-trained language representations with rich knowledge for machine reading comprehension. In *Proceedings of the 57th Annual Meeting* of the Association for Computational Linguistics, pages 2346–2357, 2019.
- [37] Wenhao Yu, Chenguang Zhu, Zaitang Li, Zhiting Hu, Qingyun Wang, Heng Ji, and Meng Jiang. A survey of knowledge-enhanced text generation. arXiv preprint arXiv:2010.04389, 2020.
- [38] Yizhe Zhang, Michel Galley, Jianfeng Gao, Zhe Gan, Xiujun Li, Chris Brockett, and Bill Dolan. Generating informative and diverse conversational responses via adversarial information maximization. arXiv preprint arXiv:1809.05972, 2018.
- [39] Tiancheng Zhao, Ran Zhao, and Maxine Eskenazi. Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 654–664, 2017.
- [40] Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. Texygen: A benchmarking platform for text generation models. In *The* 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, pages 1097–1100, 2018.



Input: Title *T* and Article *A*

Output: Entity Interaction Graph

- 1 Do word segmentation of Title T and Article A ;
- 2 Do Named Entity Recognition(NER) and keywords extraction algorithm of Article A and get the entity set E;

3 while not at end of this article do

- 4 read current sentence s;
- 5 **if** s contains $e \in E$ then
- 6 Add s to node n_e ;
- 7 else
- 8 Add s to node n_{empty} ;

9 end

```
10 end
```

11 Assign Title T as node n_t ;

- 12 **for** node n_i and n_j **do**
- 13 Edge Weight w_{ij} = number of shared sentences of n_i and n_j

14 end



(a) Entities from KG.

(b) Entities from articles.

Figure 4: The sources of entities in generated queries by different models, where KEDY generates queries with more entities from KG.

A PROBLEM FORMULATION

Given a news article $A = \{s_1, s_2, ..., s_n\}$ with s_i representing each sentence in the article and an external knowledge graph K, query generation aims to generate a set of fluent, diversified and attractive queries that are semantically relevant to A, where each query $Q = (q_1, q_2, ..., q_m)$ is a sequence of tokens. A knowledge graph contains entities as well as their related facts which are usually organized as a set of triples with each triple represented as < s, p, o >. To leverage the knowledge we need, we construct a central graph \mathbb{G}_C and a multi-hop graph \mathbb{G}_M extracted from the knowledge graph K.

The encoder represents the article *A* as a representation set $H = {\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_n}$, the central graph \mathbb{G}_C as a representation set

 $G = \{\mathbf{g}_1, \mathbf{g}_2, ..., \mathbf{g}_l\}$ and the multi-hop graph \mathbb{G}_M as a representation set $P = \{\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_k\}$.

The decoder generates the *t*-th word in the query according to the previous t - 1 generated words $q_{< t} = \{q_1, q_2, ..., q_{t-1}\}$, the article *A*, the central graph \mathbb{G}_C and the multi-hop graph \mathbb{G}_M :

$$P(Q|A, \mathbb{G}_C, \mathbb{G}_M) = \prod_{t=1}^m P(q_t|q_{< t}, A, \mathbb{G}_C, \mathbb{G}_M)$$
(14)

Then it minimizes the cross-entropy loss L and optimizes all parameters end-to-end:

$$L = \sum_{t=1}^{m} CrossEntropy(q_t^*, q_t)$$
(15)

where q_t^* is the token from the golden query.

B DETAILS OF OUR APPROACH

This part presents the details of our approaches including article graph construction and central graph encoding.

B.1 Article Graph Construction

To fully understand the long news article, we employ a graph structure modeling news article. The article graph construction described in Section 3.1.1 is displayed in Algorithm 1.

B.2 Details of Central Graph Encoding

In Section 3.2.2, $f_r^{e_j -> e_i}(\mathbf{g}_{e_j}^{l-1})$ aggregates the semantic information of each neighbor entity e_j with relation r:

$$f_r^{e_j \to e_i}(\mathbf{g}_{e_j}^{l-1}) = \boldsymbol{\alpha}_r^{e_j} FFN(\mathbf{r} \circ \mathbf{g}_{e_i}^{l-1})$$
(16)

where \circ is a concatenation operator and **r** is the relation embedding of *r*.

The attention weight is calculated by the relation embedding **r**, the article representation \mathbf{h}_{l-1} and the PageRank score controls propagation of embeddings along paths starting from e_i :

$$\boldsymbol{\alpha}_{r}^{e_{j}} = softmax(\mathbf{r} \cdot \mathbf{h}^{l-1}) \cdot PageRank(\mathbf{g}_{e_{j}}^{l-1})$$
(17)

B.3 Details of Context Representation

The attention weights in the context representations of the article, the central graph and the multi-hop graph are calculated as follows:

$$\alpha_{t-1}^{i} = \sigma(\mathbf{s}_{t-1} \cdot \mathbf{h}_{i})$$

$$\beta_{t-1}^{e_{i}} = \sigma(\mathbf{s}_{t-1} \cdot \mathbf{g}_{e_{i}})$$
(18)

$$\gamma_{t-1}^{p} = \sigma(\mathbf{s}_{t-1} \cdot \mathbf{p}_{e_p})$$

where \mathbf{h}_i is the hidden embedding of each node in the article graph. \mathbf{g}_{e_i} is the central graph representation of $e_i \in \mathbb{G}_{\mathbb{C}}$ and \mathbf{p}_{e_p} is the multi-hop graph representation of $e_p \in V^1$ aggregating two-hop neighbor entities e_q .

C MORE DETAILS ABOUT EVALUATION

C.1 Human Evaluation metrics

Correlation, the relationship between the queries and the article. Since query generation is not the same as summarization and title generation, queries related to a specific point mentioned in the article can also be thought high correlation, not necessarily the summary of the article. **Diversity**, which mainly focuses on the differences between the queries generated in test outputs. **Informativeness**, which measures whether queries contain enough information. **Fluency**, which evaluates if the phrase is fluent and complies with grammar, logic rules, and people's perception. **Novelty**, which measures whether generated queries are attractive to users to click on them. We use Spearman's rank score to measure

the correlation between raters. The Spearman's rank of around 0.7 shows a good correlation between the raters.

C.2 Supplementary Results for Diversity Evaluation

This section provides the supplementary results of diversity evaluation. The integration of knowledge greatly improves the diversity of queries.

To evaluate the effect of knowledge incorporation, we count the sources of the entities in the generated queries for all the test outputs on both the entertainment and the sport dataset. We generate 5 queries for each news article. From Fig. 4, we find that in the queries generated by our model KEDY, quite a few entities come from the knowledge graph.