Towards Explainable Search Results: A Listwise Explanation Generator

Puxuan Yu
University of Massachusetts Amherst
Amherst, MA, USA
pxyu@cs.umass.edu

Razieh Rahimi
University of Massachusetts Amherst
Amherst, MA, USA
rahimi@cs.umass.edu

James Allan
University of Massachusetts Amherst
Amherst, MA, USA
allan@cs.umass.edu

ABSTRACT

It has been shown that the interpretability of search results is enhanced when query aspects covered by documents are explicitly provided. However, existing work on aspect-oriented explanation of search results explains each document independently. These explanations thus cannot describe the differences between documents. This issue is also true for existing models on query aspect generation. Furthermore, these models provide a single query aspect for each document, even though documents often cover multiple query aspects. To overcome these limitations, we propose LiEGe, an approach that jointly explains all documents in a search result list. LiEGe provides semantic representations at two levels of granularity – documents and their tokens – using different interaction signals including cross-document interactions. These allow listwise modeling of a search result list as well as the generation of coherent explanations for documents. To appropriately explain documents that cover multiple query aspects, we introduce two settings for search result explanation: comprehensive and novelty explanation generation. LiEGe is trained and evaluated for both settings. We evaluate LiEGe on datasets built from Wikipedia and real query logs of the Bing search engine. Our experimental results demonstrate that LiEGe outperforms all baselines, with improvements that are substantial and statistically significant.

CCS CONCEPTS

• **Information systems** → **Query intent**; **Information retrieval diversity**; **Novelty in information retrieval**.

KEYWORDS

Explainable search; Query aspects; Novelty and diversity

ACM Reference Format:
Puxuan Yu, Razieh Rahimi, and James Allan. 2022. Towards Explainable Search Results: A Listwise Explanation Generator. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '22), July 11–15, 2022, Madrid, Spain.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3477495.3532067

SIGIR '22, July 11–15, 2022, Madrid, Spain

ACM ISBN 978-1-4503-8732-3/22/07…$15.00
https://doi.org/10.1145/3477495.3532067

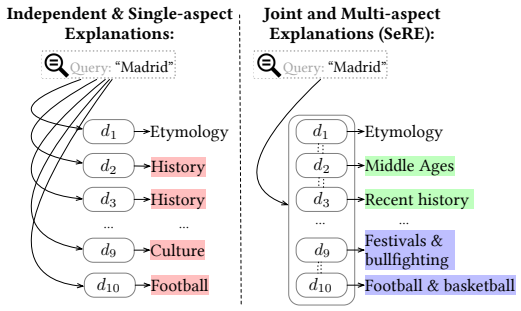


Figure 1: Explaining a list of documents in different settings.

1 INTRODUCTION

Search queries are often short and under-specified [25], encouraging the use of search result diversification techniques [59] to cover different possible query intents in the top-ranked results. To help the user understand why documents are retrieved, the list commonly presents details such as document title, URL, and a snippet – a 2- or 3-line query-focused summary of the document. Unfortunately, Thomas et al. [66] observed that less than 1% of users were able to understand the topical diversity in the search results based on such representation, suggesting huge potential value if we can improve **Se**arch **R**esult **E**xplanation (the SeRE task) to that end.

In an effort in this direction, Rahimi et al. [56] recently proposed a text generation model for aspect-oriented explanation of documents in the search results. They showed that aspect-oriented explanations help users interpret a ranked list of search results more effectively. They observed that such explanations lead to an improvement of the inter-annotator agreement on document relevance by 37% and also decrease the average time to find relevant documents by 22%. Other work has also shown the utility of aspect-oriented explanation through user studies [18, 24], demonstrating that displaying aspect coverage of documents significantly helps users efficiently locate relevant information.

However, a weakness of all those approaches to explanation is that each document in a search result list is explained individually, without regard to the other documents or where they appear in the ordering. The explanations for documents may thus be identical or at least strongly correlated – after all, they were each responses to the same query – and thus fail to help users distinguish between them. Figure 1 shows an example of the potential issue of individual explanation of documents in a ranked list. Documents d_2 and d_3 are explained with the same query aspect: correct but not helpful.

We propose LiEGe (**Li**stwise **E**xplanation **Ge**nerator) that *jointly* generates aspect-oriented explanations for *all* documents in a search

result list. That is, LiEGe creates explanations that are aware of the entire list, allowing it to highlight their differences using distinct and finer-grained query aspects as shown on Figure 1's right.

It is also well known that many documents cover multiple query aspects [8], something that is often overlooked in existing work on query aspect generation [20, 56]. Figure 1 shows that providing only one aspect per document can result in vague/under-specified explanations (d_9) and incomplete explanations (d_{10}). To address this unexplored challenge in the SeRE task, we introduce two generation strategies for *multi-aspect* documents under the joint explanation setting: (1) **comprehensive explanation generation** (CEG), where all query aspects covered by each document in the ranked list are considered as explanations; and (2) **novelty explanation generation** (NEG), where the explanation for each document describes the novel relevant information of the document with respect to the documents preceding it in the list. One type of explanation may be more suitable than the other for a particular search task or search device.

The SeRE task, then, is to generate terse relevance explanations for each document in a search result list, where explanations (1) describe the query aspects at the level of phrase, (2) are generated only from documents' contents, and (3) are diverse. Although SeRE is defined to explain search results based on query aspects, existing models for aspect mining [4, 14, 20, 26, 41, 71] or document summarization [12, 39, 60] are unfit for the problem. For example, generating query aspects and assigning them to documents does not provide diverse explanations. Section 2 provides a detailed comparison with models for these two tasks. In SeRE, phrase-level explanations are generated for results from a black-box ranker, which are different from word-based explanations through behavior approximation of black-box rankers [61, 69].

LiEGe utilizes a novel Transformer-based encoder-decoder architecture [68]. It provides multi-granular semantic representations for documents and their tokens in the whole context of search results, using different interaction signals. Document embeddings allow the encoder to leverage cross-document interactions to model their differences. More fine-grained token embeddings allow the decoder to generate distinct yet coherent explanation for each document. To achieve these representations, the encoder is a stack of the local and the proposed global encoding layers. A local layer exploits interactions between a document's tokens and the query. A global layer then models the interactions between documents in the search results. On the decoder side, we propose a cross-document attention sub-layer, which allows the decoder to attend to the entire search result before attending to the tokens within a single document to generate its explanation. With our modifications, the explanation of each document with respect to the query can appropriately reference other documents in the search result. The provided explanations describe the relevance of each document to the query and the topical diversity of all results.

To train LiEGe for the CEG and NEG settings, we constructed two weakly-labeled datasets for multi-aspect explanation from the English Wikipedia. To evaluate the explanation of relevance and topical diversity of Web search results, we adapt the MIMICS dataset [78], which is built from real query logs of the Microsoft Bing search engine, to the SeRE task. We perform extensive experiments to evaluate LiEGe and competitive baselines under different settings of SeRE (CEG and NEG). LiEGe outperforms all the baseline models significantly over both datasets and demonstrates superior transfer performance from Wiki to MIMICS. In terms of the BLEU metric, LiEGe outperforms the strongest generation baseline BART [34] by 27.2% and 27.1% on the Wiki and MIMICS datasets, respectively.

2 RELATED WORK AND BACKGROUND

2.1 Explainable Information Retrieval

Efforts toward interpreting ranking models can be categorized into the development of intrinsically interpretable models and black-box explanations. The former aims at adopting interpretable models such as additive models [82] to ranking. Black-box (or model-agnostic) models generate explanations for rankers without accessing their internal structure. The existing ranker explainers are mostly based on the Local Interpretable Model-agnostic Explanations (LIME) [58] for explanation of classification and regression models. For example, some studies [61, 69] generate word-based explanations for pointwise ranking models. Such an explanation indicates the importance of individual words in a ranker's decision regarding document relevance. Instead of using LIME, Singh and Anand [62] approximate complex rankers with simple interpretable rankers (e.g., relevance-based language model [33]) for the instance to be explained by expanding the query. Almost all existing explanations of learning-to-ranking models provide insights into their ranking behavior based on axioms [57, 70], explanation features such as individual terms [61, 62, 69], or human-engineered ranking features [63]. These explanations are mostly valuable for developers of ranking models to understand their systems, and are of limited help to users in understanding document relevance [56]. In comparison, the SeRE task, as defined by Rahimi et al. [56] and extended here, is *user-oriented* – helping users interpret and understand the documents in a search result list efficiently.

2.2 Query Aspects

Solving the SeRE task is an effort to help interpret and understand provided search results, as queries are often under-specified or ambiguous [59]. A considerable number of studies in the field of information retrieval have been devoted to address issues related to the ambiguous nature of search queries.

In one direction, several approaches have been developed for mining query aspects, where they mostly use external resources such as query logs [4, 28, 54, 72, 78], anchor text [14, 32], knowledge bases [6, 26], or a mixture of them [15, 22]. Just a few studies have been conducted to identify query aspects from only documents. These works extract query aspects from the entire target corpus [5] or first-stage retrieved documents (e.g., top-1k and more) [71]. They thus utilize many more documents than our model to extract query aspects. More related to the SeRE task, Kong and Allan have studied query facet extraction from search results [29–31]. An extracted facet is represented by a set of coordinate terms (e.g., {*AA, Delta, JetBlue*}), while we aim to directly generate the underlying concept of extracted facets/aspects, such as *airlines* for the given example.

More recently, MacAvaney et al. [41] proposed using casual language model T5 [55] to generate query variants *independent* of retrieved documents. Thus, the generated query variants cannot be used directly for explaining document relevance. Hashemi et al.

[20] introduced NMIR, which is a BART-based [34] model for query intent generation from top-ranked documents. Those documents are first clustered using the K-means algorithm [19] and then a short intent description is generated for each cluster. NMIR is not suitable to solve the SeRE task, because (1) it assumes that documents cover a single query aspect, and (2) documents within the same cluster share the same intent (explanation). Thus, NMIR does not reveal the differences between the documents within the same cluster.

Providing information related to query aspects has been shown to improve users' information seeking experience, through visualizing the document relevance degree to each query aspect [24], or by providing aspect-oriented explanation in textual format [56]. Based on the proven utility of aspect information for users through user studies [24, 56, 66], this work focuses on improving the quality of generating aspect-oriented explanations for search results. The choice of interface to present the obtained explanations is out of the scope of this work.

Tasks beyond SeRE can potentially benefit from the improved quality of generated query aspects, such as coverage-based search result diversification [27, 38, 53], query suggestion [1, 10], or clarifying question selection and generation [2, 77, 79]. Further exploration in those directions is left for future work.

2.3 Summarization and Snippet Generation

Document summarization [75], especially abstractive document summarization [65], is related to the SeRE task, as explanations should provide a concise summary of document information that is relevant to a query. Snippet generation for search results [3, 11, 12], which has been considered as a query-biased document summarization task [45], is also related to our task. There are, however, key differences between those tasks and SeRE that make the existing works on document summarization or snippet generation unfit for SeRE. First, our desired explanation for each document provides noun-phrase descriptions of relevance. These outputs are much shorter than those in popular summarization datasets [13, 23, 43, 44] and search snippets where a summary is usually one or more sentences. Second, our task requires a *many-to-many* sequence-to-sequence (seq2seq) model to jointly generate explanations for all documents in a search result list. This scheme is different from *one-to-one* seq2seq task considered in single document summarization [60] and existing models for snippet generation. It is also different from *many-to-one* seq2seq models for multi-document summarization [39, 50, 73].

Guided summarization is also conceptually related to our task. It takes a document and a guidance signal as input [16, 35, 52]. Comparing to these models, our task is more complex as there is more than one source of guidance signals: the search query and multiple other documents in the search result.

2.4 Listwise and Setwise Ranking Models

Modeling cross-document interactions has been shown to be useful for learning-to-rank [47, 49, 76] and diversification-aware ranking models [64, 74]. LiEGe exploits cross-document interactions differently from existing ranking models. First, our model is based on contextual representations of documents that are fine-tuned for the end-task, instead of a document representation based on handcrafted features [47, 49] or Doc2Vec [64, 74] that is not updated during training. Secondly, our task requires a fine-grained encoding of each document with respect to other documents in the search results. Specifically, in addition to a dense representation of each document considering its interactions with other documents, our task requires document *tokens* to be encoded in the context of other documents. This fine-grained encoding is needed for generation of natural language explanations. LiEGe also provides a solution to incorporate cross-document interactions in the *decoding* stage, which is unexplored in prior works as those list- or set-wise neural ranking models use only encoders.

2.5 Transformer

We briefly overview the Transformer [68] architecture that is needed to describe LiEGe. Transformer is based on the encoder-decoder structure, consisting of stacked encoder and decoder layers. The encoder layer in Transformer includes self-attention and position-wise feed-forward sub-layers. The decoder has an additional attention layer to utilize the encoder representations of input.

The self-attention sub-layer updates token representations using the Attn() function based on scaled dot product:

$$\text{Attn}(Q, K, V) = \text{softmax}\left(\frac{QK^\intercal}{\sqrt{m}}\right)V, \tag{1}$$

where $Q \in \mathbb{R}^{N_Q \times m}$ denotes the query matrix, and $K, V \in \mathbb{R}^{N_K \times m}$ are the key and value matrix. N_Q is the query size[1], N_K is the key/value size, and m is the dimension of hidden vectors. In the case of multi-head attention, three inputs Q, K, and V are projected to h different sub-spaces of the new dimension $\hat{m} = m/h$. The attention is performed in each sub-space with $\text{Attn}(\cdot,\cdot,\cdot)$ in Eq. 1, and the outputs from all sub-spaces are gathered as:

$$\text{MHA}(Q, K, V) = \text{Concat}\left(\left[\text{Attn}(QW_i^Q, KW_i^K, VW_i^V)\right]_1^h\right), \tag{2}$$

where $W_i^Q, W_i^K, W_i^V \in \mathbb{R}^{m \times \hat{m}}$ are projection weights in the i-th subspace. In practice, input matrices have an additional dimension for batching. Denoting batch size with b, inputs are of dimensions:

$$Q \in \mathbb{R}^{b \times N_Q \times m}, \quad K, V \in \mathbb{R}^{b \times N_K \times m}, \tag{3}$$

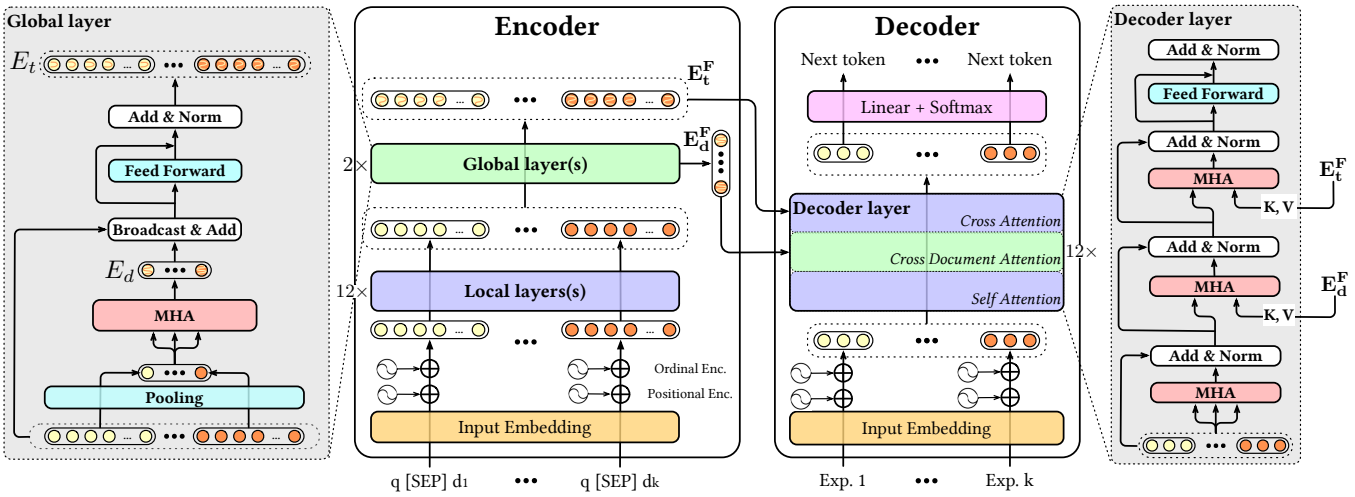
Note that $\text{MHA}(Q, K, V)$ always shares the same shape as input Q.

3 LIEGE MODEL

Given a query q and the top-ranked documents $R = \{d_1, \ldots, d_k\}$ retrieved by a black-box ranker, the goal of LiEGe is to generate an explanation for each document describing what information each document in R provides with respect to the query q. Explanations provide the query aspect(s) that are covered by each document d_i in the proposed **novelty-** or **comprehensive**-based manner.

In order to describe the relevant and distinct information that each document in a search result list provides, a document needs to be encoded with respect to the query and with respect to the other documents in the list. The former is required to capture the relevant part of a retrieved document as a small portion of a document may be all that is related to the query [67]. The latter part of encoding

[1] "Query" here is different from an input search query q.



(a) Details of global layer. (b) Model overview. (c) Details of decoder layer.

Figure 2: The architecture of LiEGe.

exploits cross-document interactions so that generated explanations can help users distinguish the differences between documents in the search result. The encoded input is then passed to the decoder to generate explanations. Figure 2 (b) shows the architecture of LiEGe. In the following we provide a detailed description of the encoder and decoder, starting with how an input sequence to the model is built.

3.1 Input Representation

Given query q and retrieved documents $R = \{d_1, \ldots, d_k\}$, the query is concatenated with each document with a separator token in between. Each concatenated $q - d_i$ sequence is tokenized and then truncated or padded up to the max length of l tokens. The positional and segment embeddings are added to the input token embeddings. For generating novelty explanations, the model requires the document ranks in a search result list so that it can detect document novelty regarding its preceding documents. For this reason, we add *ordinal encodings* to the input embeddings. The ordinal encodings have the same dimension as m and are learned using an Embedding function P, similar to the one used by Pang et al. [47]. The rank of document d_i is passed to the function P that encodes the absolute ranking position into an embedding vector $p_i \in \mathbb{R}^m$. The p_i embedding is then added to every token embedding of document d_i. Finally, the query-document pair (q, d_i) is represented by $X_i \in \mathbb{R}^{l \times m}$, and the entire list R is represented by $X = [X_1, \ldots, X_k] \in \mathbb{R}^{k \times l \times m}$.

3.2 Encoder

The encoder consists of a stack of local and global attention layers. A local attention layer updates token representations based on intra-document self-attention, while a global attention layer updates representations based on inter-document attention. In other words, local layers perform per document computation while global layers perform per result list computation.

We denote the input *token* representations to the h-th transformer layer of the encoder as $E_t^{(h)} \in \mathbb{R}^{k \times l \times m}$. Note that $E_t^{(1)} = X$ defined above. The output token representations of the layer are denoted by $E_t^{(h+1)} \in \mathbb{R}^{k \times l \times m}$, which constitute the input to the next layer if any. A global layer has one additional output compared to a local one. The additional output is *document* representations for all documents in the result list, denoted by $E_d^{(h+1)} \in \mathbb{R}^{k \times m}$. These document representations will also be used in the decoder.

Local layers perform multi-head self attention on the token sequence from a query-document pair (intra-document), similar to the encoder layers in Transformer [68]. The contextualized representations are then linearly transformed. Residual connection and layer normalization are applied for each of the layers [68]. The function of a local layer can be formalized as:

$$L_t = \mathrm{LN}(E_t^{(h)} + \mathrm{MHA}(E_t^{(h)}, E_t^{(h)}, E_t^{(h)})), \tag{4}$$

$$E_t^{(h+1)} = \mathrm{LN}(L_t + \mathrm{FFN}(L_t)), \tag{5}$$

where $\mathrm{LN}(\cdot)$ is layer normalization, and $\mathrm{FFN}(\cdot)$ stands for position-wise feed-forward networks.

Global layer first generates dense document embeddings by pooling, where each document in the search result is represented with a single embedding of dimension m. We consider two pooling strategies: (1) applying multi-head pooling [39] to learn a weighted average of the embeddings of the document's tokens; and (2) simply taking the embedding of the first token ([CLS]) as the representation of the entire sequence. After acquiring a list of dense embeddings for documents via pooling, multi-head self-attention is applied across documents in the search result list. The output is still a list of dense embeddings, where each document embedding is contextualized based on the other documents in the search result. In order to propagate information from document interactions at the document granularity to the token granularity, the contextualized embedding of a document is added to the embedding of each of its tokens. We

refer to this function as *broadcast & add*. Figure 2 (a) shows the architecture of a global layer. Specifically, the outputs of a global layer are calculated as:

$$E_d^{(h)} = \text{Pool}(E_t^{(h)}), \tag{6}$$

$$E_d^{(h+1)} = \text{MHA}(E_d^{(h)}, E_d^{(h)}, E_d^{(h)}), \tag{7}$$

$$\hat{E}_t[i,j] = E_t^{(h)}[i,j] + E_d^{(h+1)}[i], \tag{8}$$

$$E_t^{(h+1)} = \text{LN}(\hat{E}_t + \text{FFN}(\hat{E}_t)), \tag{9}$$

where $E_t^{(h)}[i,j]$ is the embedding of the j-th token of document i in the input token representations $E_t^{(h)}$. $E_d^{(h+1)}[i] \in \mathbb{R}^m$ is the representation of the i-th document contextualized based on all documents in the search result. Eq. 8 shows the broadcast & add operation, which is performed for all tokens of all documents. Therefore, information from all documents is considered and appropriately reflected in the representation of every document token.

Encoder outputs. We denote the output token representations from the *final* encoder layer as E_t^F, and the contextualized document representations from the *final global* layer in the encoder as E_d^F. Note that E_t^F can be the output of a global or a local transformer layer, depending on the model composition. E_t^F and E_d^F are used as inputs to the decoder.

3.3 Decoder

Each layer in the decoder of the Transformer [68] contains two attention sub-layers: a self-attention sub-layer and a cross-attention (also called "encoder-decoder attention") sub-layer. The purpose of self-attention in the decoder is to effectively use the already generated text for the prediction of the next token. The decoder uses cross-attention to utilize the encoder representations of input for identifying which part of the input sequence it should focus on to predict the next token. To avoid confusion, we refer to this sub-layer as **cross-token** attention.

Cross-document attention sub-layer. We propose a cross-document attention sub-layer, which is placed between self-attention and cross-token sub-layers in each decoder layer. The details of a decoder layer in LiEGe are depicted in Figure 2 (c). With cross-document attention, the representation of each token generated so far is updated by information from the contextualized embeddings of all documents in the search result from the encoder (E_d^F). Attention to global information E_d^F helps the model to identify which specific query aspect should be generated. Then, through attention to local document information (E_t^F), the model identifies aspect-related part(s) of the document to be used for generation of the next token.

Input to the h-th layer of the decoder is denoted as $D_t^{(h)} \in \mathbb{R}^{k \times l' \times m}$, where l' is the maximum output length (during training) or the length of currently generated sequence (during inference). The function of a decoder layer with a cross-document attention sub-layer is formally formulated as follows.

$$D_t = \text{LN}(D_t^{(h)} + \text{MHA}(D_t^{(h)}, D_t^{(h)}, D_t^{(h)})), \tag{10}$$

$$D_t = \text{LN}(D_t + \text{MHA}(D_t, E_d^F, E_d^F)), \tag{11}$$

$$D_t = \text{LN}(D_t + \text{MHA}(D_t, E_t^F, E_t^F)), \tag{12}$$

$$D_t^{(h+1)} = \text{LN}(D_t + \text{FFN}(D_t)). \tag{13}$$

After the final decoder layer, a linear and a softmax layer predict the next token to be generated for the explanation of each document. At inference time, generation repeats until either the end-of-sentence token is generated or the maximum output length is reached.

3.4 Training of LiEGe

Batching. Instead of random query-document pairs for mini-batch training in Eq. 3, we need to group documents from the same search result together to leverage their interactions. We define *group size* k as the maximum number of documents considered in a search result for a query. During training, search results with less than k documents are padded to the constant list size k. We then define *batch size* b as the number of groups (SERPs) included in a batch. In other words, $b \times k$ documents are input into the model in one batch, with b groups running in parallel. We use the global document mask to make sure that (1) padded documents are masked in local and global attention calculation; and (2) global attention is restricted to documents within the same group. In the experiments, we consider up to 10 documents per query (i.e., $k = 10$), which is similar to the first page returned by modern search engines.

Learning objectives. Similar to prior works on sequence-to-sequence transduction, we use the cross-entropy of predicted and gold probability distribution at each position [34] as the loss function to guide parameter optimization. The loss function is computed over all positions in sequences to be generated.

4 DATASETS AND EVALUATION METRICS

We evaluate LiEGe for two different types of explanation strategies to fully demonstrate the advantages of listwise modeling in content-based explanation generation: comprehensive explanation generation (CEG) and novelty explanation generation (NEG). Each type of explanation requires its own training and test set. In practice, these two tasks take the same set of inputs (a query and a list of documents), but have different outputs. To the best of our knowledge, there is no public dataset for listwise content-based explanation of search results. We thus adapt existing datasets for other similar tasks to the SeRE task. In the following, we describe how those datasets are built and processed. The processed datasets used in this work are made public on https://github.com/PxYu/LiEGe-SIGIR2022.

4.1 Wikipedia as a Weakly Labeled Dataset

One solution to automatically build a large-scale training dataset is to consider a Wikipedia article as a search result: the title of the article (mostly an entity name) resembles the query, the content of each section of the article is considered as a document retrieved for the query, and the section heading is an aspect-based explanation of how that section is related to the query and to other sections. Note that most sections (documents in our analogy with a search result list) in Wikipedia cover a single aspect of the query. This is because Wikipedia articles have been topically organized into sections through various iterations by human experts. We thus refer to this dataset from Wikipedia as **Wiki-SA**, where SA stands

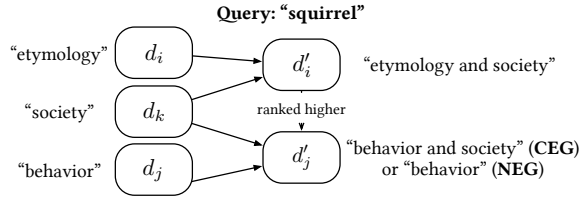


Figure 3: An example of creating multi-aspect documents.

for single-aspect. This dataset is similar to the Wiki dataset used by Rahimi et al. [56] and to the WikiOG dataset [80].

The Wiki-SA dataset is not suitable for training and evaluation of explanation generation for documents covering multiple query aspects. ClueWeb and MS MARCO, used for pointwise explanation [56], are not suitable either. This is because there are very limited or almost no documents with multiple annotated query aspects in ClueWeb and MS MARCO, respectively.

In order to build a large amount of training data from Wikipedia for the CEG and NEG settings, we propose a simple approach to intentionally create overlapping contents between documents in a search result, such that some documents cover multiple aspects of the query. We call this process "**fusing**" with an example shown in Figure 3. Given documents from the same Wikipedia article, we randomly sample three documents d_i, d_k, and d_j, without replacement, whose original aspect-based explanations are e_i, e_k and e_j, respectively. Using d_k to create overlapping content, we concatenate d_i and d_k into d'_i, and concatenate d_j and d_k into d'_j. Documents are concatenated in random order to make sure that an explanation model is able to detect document novelty, instead of memorizing the position of novel information in documents. The new documents d'_i and d'_j are used to create two datasets referred to as **Wiki-CEG** and **Wiki-NEG**. For Wiki-CEG, the explanations for d'_i and d'_j are labeled as "e_i & e_k" and "e_j & e_k", respectively. To build ground-truth explanations for Wiki-NEG, the ranks of documents in a search result list are needed. We assume, without loss of generality, that d'_i is ranked higher than d'_j in the result list. Then d'_i and d'_j are explained with "e_i & e_k" and 'e_j", respectively. Although d'_j covers e_i, this aspect is already covered by the preceding document d'_i. The fusing process creates two documents from three sections in a Wikipedia article. From a Wikipedia article with s sections, we thus build $\lfloor s/3 \rfloor \times 2$ documents. If there are any sections not used by fusing (there will be at most two such sections), those sections will be kept in the search result even though they each cover a single query aspect.

All three datasets Wiki-SA, Wiki-CEG, and Wiki-NEG are built from the same set of Wikipedia articles. We select Wikipedia articles with at least six sections of 128-256 words apiece. The reason for a minimum of 128 words in a section is to have documents with enough content. The max section length is set to due to the maximum length of 512 tokens as the input of BERT and the concatenation of sections in construction of Wiki-CEG and Wiki-NEG. The last constraint of six sections makes sure that after fusing, a search result list contains at least four documents.

Filtering with the above constraints, 39,287 Wikipedia articles remained. Each article constitutes an instance of a query and search result list (q, R) in the Wiki-SA, Wiki-CEG, and Wiki-NEG datasets. The instances are split into train/dev/test in the 80%/10%/10% ratio. The dev set is used for tuning hyper-parameters and early stopping of training. The Wiki-CEG and Wiki-NEG datasets differ from Wiki-SA in terms of the number of documents in search results and the document length because of fusing. Table 1 reports the statistics of Wiki datasets: the number of instances, the average number of documents in search results, the average length of documents, and the average length of explanations.

Table 1: Statistics of created datasets.

Dataset	Wiki			MIMICS	
	-SA	-CEG	-NEG	-CEG	-NEG
#query	39,287	39,287	39,287	1,992	1,992
#docs-per-query	7.5	5.4	5.4	5.2	3.0
#words-per-doc	184.9	344.2	344.2	54.0	54.0
#words-per-explanation	2.7	5.1	3.9	1.9	1.4

4.2 MIMICS Dataset

MIMICS [78] is a collection of datasets for search clarification built from real search queries sampled from the Bing query logs. Besides its real queries and search results, MIMICS has another advantage over the Wiki datasets: two documents in a search result list can cover the same query aspect without having exactly the same content for the common aspect. This property makes the evaluation of CEG and NEG more realistic.

Each clarification in MIMICS consists of a query, a clarifying question, up to five candidate answers for the clarifying question which are aspects of the query, and the top-10 documents retrieved by Bing. For SeRE, we need to have gold explanations for documents based on query aspects. However, MIMICS does not contain aspect-level relevance information. In other words, the top-10 documents retrieved w.r.t. a query as well as the query aspects are provided, but which documents are relevant to which query aspects are not specified. To adapt MIMICS for the SeRE task, we make the conservative assumption that a document is considered relevant to a query aspect only if it contains the aspect terms. We thus obtain high quality labels for aspect-level relevance, at the cost of missing some relevance labels.

We chose the ClickExplore version of MIMICS as it contains the largest number of unique queries. We perform the following processing steps. (1) Query terms are removed from aspects as repeating query terms in explanations does not provide additional information; (2) The concatenation of a document's heading and snippet is used as the document content [20]. Note that full document contents are not released in the MIMICS dataset; (3) Documents that are not labeled as relevant to any query aspects are removed; (4) Query aspects that are not associated with any documents in a search result list are also removed. If the number of remaining query aspects is less than three, the query is removed; (5) Queries whose clarification has engagement level below 4 (out of 10) are removed. Engagement level indicates the quality of clarification and query aspects perceived by users [78]. In the end, we acquired 1,992 queries from the original dataset, which are split into train/test set evenly. Similar to the Wiki datasets, we create two variants from MIMICS to separately evaluate CEG and NEG. For **MIMICS-CEG**,

Table 2: Results for comprehensive explanation generation, evaluated on the Wiki-SA and Wiki-CEG datasets.

Dataset	Wiki-SA (single-aspect)						Wiki-CEG (multi-aspect)					
Metric	BLEU	B-1	R-1	R-L	BERTScore	Div (↓)	BLEU	B-1	R-1	R-L	BERTScore	Div (↓)
TextRank	0.12	12.58	15.74	13.84	37.97	62.02	0.11	16.59	14.91	12.73	42.35	70.33
TS-TextRank	0.23	11.19	13.91	12.21	37.65	62.75	0.26	14.01	12.47	10.74	41.83	71.08
BERT-LIME	0.30	6.50	8.31	7.93	42.06	60.07	0.16	7.10	8.63	8.18	43.58	68.56
KeyBERT	2.11	13.81	16.87	16.40	46.04	58.43	1.52	15.11	12.98	12.51	49.90	66.98
GenEx	6.55	25.56	21.06	20.94	54.85	53.20	0.30	11.72	6.41	6.37	47.16	69.72
HiStGen	8.86	22.59	17.49	17.23	53.92	48.29	11.73	42.47	39.30	34.90	64.10	55.82
BERT	17.28	40.14	39.71	39.45	65.60	45.29	15.61	48.86	46.54	41.28	67.77	53.46
LiEGe (BERT)	19.27	41.40	40.76	40.55	65.98	45.03	17.36	50.46	47.69	42.52	68.40	53.00
BART	18.51	42.13	42.16	41.90	67.03	45.23	16.53	49.54	47.42	42.23	68.33	53.42
LiEGe	**21.97***	**45.56***	**45.84***	**45.65***	**69.01***	**44.47***	**18.79***	**51.72***	**49.37***	**44.32***	**69.46***	**52.73***

the explanation of a document contains all associated query aspects. For **MIMICS-NEG**, a document's explanation only contains aspects that are novel considering its preceding documents. In cases that a document contains no novel aspects, we discard it. Statistics of these datasets are also reported in Table 1.

4.3 Evaluation Metrics

To evaluate the quality of generated explanations, we mainly use BLEU F_1 [48] (using sacreBLEU [51]) which is a standard metric for evaluation of natural language explanations [37] and text generation models [34, 56, 68]. We report BLEU-1 as B-1 and the weighted geometric mean of BLEU-k (k=1,2,3,4) as BLEU. We also report ROUGE-1 F_1 and ROUGE-L F_1 [36] as R-1 and R-L, respectively. For evaluation of multi-aspect explanations, we consider different aspects as multiple references in computation of the BLEU and ROUGE metrics. To measure the *semantic* similarity between generated and ground-truth explanations, we use BERTScore [81]. We report micro-averaged BLEU, ROUGE, and BERTScore on all document-explanation pairs in a test set.

In addition to accuracy-based metrics, we measure the diversity of generated explanations for a search result list, referred to as *Div*. For this purpose, we compute the average semantic similarity of all pairs of explanations that are generated for a search result list. The semantic similarity of explanations is computed using BERTScore. A lower Div score indicates more diversity in the explanations of a search result list.

We use t-test with Bonferroni correction for statistical significance test at the level of 95%. Statistical significant improvements of LiEGe over *all* baselines are marked with * in the result tables.

5 EXPERIMENTAL SETTINGS AND RESULTS

5.1 Compared Models

To the best of our knowledge, there is no model for listwise content-based explanation of search results. For evaluation of LiEGe, we thus adapt some representative models for tasks similar to SeRE and report their performance. The compared models are as follows.

TextRank [42] is an unsupervised model for extraction of document keywords using the PageRank algorithm.

TS-TextRank is a modified version of TextRank based on *topic-sensitive* PageRank [21] where query terms are used as the topic [56].

BERT-LIME uses LIME [58] to explain the basic BERT-based ranker as fine-tuned by Nogueira and Cho [46]. This is done similar to the EXS model [61] except that LIME is applied on the real-valued (perturbed) document scores, instead of using heuristics to convert document scores into binary relevance labels. This difference can increase the fidelity of the explainer.

KeyBERT [17] is a pretrained BERT-based keyphrase extraction model, which uses BERT embeddings and their cosine similarity to find the phrases in a document that are the most similar to the document itself. We consider n-grams where n varies from 1 to 5, and take the top-ranked keyphrase as an explanation.

KeyBERT-MMR is a variant of KeyBERT for generating novelty explanations. It scores phrases using maximal marginal relevance [7] considering both the similarity with the document and *novelty* compared to the preceding document(s). Given document d and its preceding documents S_d, a phrase p in d is scored with

$$\lambda \, \mathrm{Sim}(p, d) - (1 - \lambda) \max_{d' \in S_d} \mathrm{Sim}(p, d'),$$

where $\mathrm{Sim}(\cdot, \cdot)$ represents cosine similarity of their BERT embeddings. We set $\lambda = 0.7$ because it yields the best performance on the dev set of Wiki-NEG.

GenEx [56] is a BERT-based model for generating an aspect-oriented explanation for *individual* documents with a noun phrase, even documents covering multiple query aspects. We report the results of GenEx trained on a dataset similar to Wiki-SA, but larger in size.

HiStGen [80] is a hierarchically structured model for detecting section boundaries and generating headings for the obtained sections. As documents in a search result list have clear boundaries, we only train and test HiStGen to generate headings as explanations. We implement this approach in PyTorch since its implementation is not publicly available.

NMIR [20] is a BART-based model for generating query intents by clustering the top search results (details in Section 2.2). We treat the generated intent description as explanations for all documents in a cluster. NMIR was trained on the full MIMICS dataset (approximately 340K SERPs; in comparison, our training/test set contains about 1K SERPs). We try different trained checkpoints of NMIR, and report the best performing results on MIMICS-CEG. NMIR is used as a baseline to demonstrate the difference between query intent generation and the SeRE task.

Table 3: Results for NEG evaluated on Wiki-NEG.

	BLEU	B-1	R-1	R-L	BERTScore	Div (↓)
KeyBERT	1.65	12.03	11.89	11.51	46.04	60.47
KeyBERT-MMR	1.77	12.90	12.83	12.30	47.14	58.62
HiStGen	8.31	29.47	25.64	23.98	56.65	51.68
BERT	10.75	34.50	31.69	29.71	59.47	49.66
LiEGe (BERT)	13.35	36.86	33.91	31.62	60.13	47.46
BART	11.84	35.94	33.80	31.84	60.82	49.38
LiEGe	**15.06***	**39.45***	**38.02***	**35.80***	**62.74***	**47.13***

BART [34] is a powerful seq2seq generation model. It has 12 local layers in the encoder and 12 decoding layers in the decoder, where the decoder layers do not have a cross-document attention sub-layer. BART explains each document in the search result individually, and thus is a pointwise explanation model. We report the performance of BART to show the effectiveness of listwise generation of explanations for search results.

LiEGe (Section 3) in its default configuration with 12 local layers followed by 2 global layers, using multi-head pooling (8 attention heads) for the dense embeddings of documents, and employing ordinal encoding.

BERT and **LiEGe (BERT)** are other baselines to provide a fair comparison with the GenEx and KeyBERT models as they are based on a pre-trained BERT. These baselines demonstrate that performance gains by LiEGe is not due to a pre-trained decoder in BART.

5.2 Details about Training and Inference

All local layers in the encoder and all decoder sub-layers except cross-document sub-layers (if any) of BART and LiEGe are initialized with weights from a pre-trained BART checkpoint[2]. LiEGe (BERT) is initialized with BERT weights[3]. For training BERT and LiEGe (BERT), we set the batch size to 20, i.e., 200 documents from 20 search result lists in one batch, evenly distributed across 4 Nvidia RTX-8000 GPUs. For BART and LiEGe, batch size is set to 8 due to the larger model size compared to BERT. Each model is trained on each training dataset, with the learning rate decreasing linearly from 5e-5 to 0. We use AdamW optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 1e-6$) with bias correction for optimization [40]. Training on datasets for the CEG and NEG settings (such as Wiki-CEG or MIMICS-CEG), multiple aspects in the gold explanations of some documents are ordered randomly. If a search result in a Wiki dataset contains more than 10 documents, we randomly select 10 of them. For generation-based models GenEx, HiStGen, NMIR, BERT, BART, and LiEGe, we use greedy search as the decoding strategy. The maximum number of tokens to be generated by the decoder is set to 32. In practice, the generated outputs are much shorter than this maximum length.

5.3 Results

Comprehensive explanation generation on Wiki. Performance of LiEGe and baseline models over the Wiki-SA and Wiki-CEG datasets are reported in Table 2. The first block in the table contains the unsupervised models TextRank and TS-TextRank, BERT-LIME, and the two BERT-based models KeyBERT and GenEx. NMIR [20] is trained on the MIMICS dataset, thus instances from the Wiki datasets are out of its distribution. Performance of NMIR on the Wiki datasets is not thus comparable to other models, and omitted here. Note that GenEx is trained on a dataset very similar to Wiki-SA. Thus, GenEx performs the best on Wiki-SA within this category. However, the performance of GenEx dropped significantly on Wiki-CEG. This is because GenEx is trained to generate one terse explanation for a document, and does not generate a list of aspects. Other baseline models rank the terms/phrases in a document and thus it is possible to select more outputs for each document, complying with our setting of SeRE. TextRank and TS-TextRank extract a set of keywords, while KeyBERT extracts coherent noun phrases. The BLEU performance of TextRank and TS-TextRank is significantly lower than that of KeyBERT, because the BLEU metric considers higher-order n-grams (B-2, B-3 and B-4) as well.

The second block only contains HiStGen. It is separated from the models in the first block in that it is trained on our datasets before testing. Note that we removed the review mechanism [9] in HiStGen on Wiki-CEG because it prevents duplicate terms among outputs for different documents in the search result, while ground-truth labels for these documents in Wiki-CEG have common query aspects, and thus common terms. In the third block, we include BERT and LiEGe (BERT). The final block contains BART and the complete version of LiEGe. In all cases, LiEGe is able to outperform its counterpart base model significantly. In addition, LiEGe substantially outperforms all the baselines where the improvements are also statistically significant. BERT and LiEGe (BERT) underperform BART and LiEGe (BART), respectively, and are thus omitted from Tables 4 and 5 due to space limit.

Improvements of LiEGe over pointwise explanation models GenEx, BERT, and BART demonstrate the importance of listwise encoding and explanation of search results. Improvements over HistGen show that LiEGe successfully leverages information from cross-document interaction in the comprehensive setting of the SeRE task.

Novelty explanation generation on Wiki. For evaluation of the *novelty* explanations, we do not compare against TextRank, TS-TextRank, LIME, and GenEx, because these models do not exploit

Table 4: Results for CEG evaluated on MIMICS-CEG.

	BLEU	B-1	R-1	R-L	BERTScore	Div (↓)
TextRank	0.12	8.62	12.03	11.36	34.68	78.86
TS-TextRank	0.19	8.27	11.76	11.23	34.70	78.45
BERT-LIME	0.05	1.47	0.29	0.29	36.65	78.28
KeyBERT	0.79	9.36	12.81	12.61	39.46	77.00
GenEx	0.35	3.92	3.23	3.23	40.45	75.93
NMIR	0.03	3.34	6.91	6.28	32.19	99.70
HiStGen	19.82	39.65	31.14	30.56	55.60	64.16
BART	41.59	59.90	51.76	51.53	68.50	59.25
LiEGe	39.25	62.85	57.02	56.50	71.06	57.73
BART (Wiki)	45.61	63.65	59.06	58.10	72.13	57.52
LiEGe (Wiki)	**49.11***	**68.91***	**64.96***	**64.09***	**76.37***	**56.34***

Table 5: Results for NEG evaluated on MIMICS-NEG.

	BLEU	B-1	R-1	R-L	BERTScore	Div (↓)
KeyBERT	0.79	7.00	10.43	10.42	37.29	75.34
KeyBERT-MMR	0.85	7.54	11.25	11.15	39.94	72.66
HiStGen	13.14	23.21	18.29	18.29	53.47	63.85
BART	25.49	44.74	42.08	41.99	64.93	55.28
LiEGe	33.17	55.94	52.83	52.67	71.26	51.39
BART (Wiki)	29.10	49.98	47.46	47.36	67.51	51.51
LiEGe (Wiki)	**37.00***	**61.02***	**59.02***	**58.87***	**75.40***	**50.68***

[2] https://huggingface.co/facebook/bart-base

[3] https://huggingface.co/bert-base-uncased

Table 6: Example generated explanations. Multiple aspects in explanations are separated by "||". The strikethrough aspects are removed from gold explanations in the NEG setting. Differences between BART and LiEGe are highlighted.

Query	Wiki-CEG: "Madrid"[4]			
Labels	d_1: Etymology || Middle Ages	d_2: Etymology || Francoist dictatorship	d_3: Location || Literature	d_4: Location || Cuisine
KeyBERT [17]	villa de realengo	substandard housing	oldest urban core	culinary specialty
NMIR [20]	madrid in spanish	madrid in spanish	madrid italian in italian	madrussian in english
HistGen [80]	history	background	location	culture
BART [34]	etymology and history	etymology and history	culture and location	culture and location
LiEGe (ours)	etymology and middle ages	franco regime and etymology	location and literature	location and the madrilenian cuisine

Query	MIMICS-NEG: "Chlortalidone"			
Labels	d_1: side effects	d_2: ~~side effects~~ || interactions	d_3: ~~side effects~~ || warnings	d_4: brand name
KeyBERT-MMR [17]	adjunctive therapy	the blood vessels	educational purposes	hormone or steroid
NMIR [20]	chlorthalidone china	chlenthalidone uk	chlenthalidone uk	charlorthalidones usa
HistGen [80]	information	medication	indications	description
BART [34]	side effects	side effects	warnings and side effects	brand
LiEGe (ours)	side effects	interactions	warnings	brand names

the contexts of other documents, and thus cannot detect document novelty. Results on the Wiki-NEG dataset are shown in Table 3. First, we observe that adding the MMR component to KeyBERT improves its performance. It promotes novelty by penalizing phrases that are more similar to those selected for the preceding documents. HiStGen can also generate novelty explanations because it incorporates a review mechanism [9]. LiEGe is the best performing model in the novelty setting of SeRE. Though listwise modeling of search results for comprehensive explanation generation is shown to be effective (Table 2), it is more critical for novelty explanations because information from preceding documents is essential. The results in Tables 3 and 2 show that LiEGe (BART) achieves higher percentages of improvements over baselines in Wiki-NEG compared to Wiki-SA and Wiki-CEG datasets.

Explanation generation on MIMICS. Performance of LiEGe and baseline models on MIMICS-CEG and MIMICS-NEG are reported in Tables 4 and 5. The tables show the results of LiEGe and BART models when they are first pre-trained on the Wiki dataset and then fine-tuned on the corresponding MIMICS dataset (marked with "Wiki"), as well as when they are just trained on a MIMICS dataset. LiEGe almost always outperforms all baselines and its counterpart base model in both CEG and NEG settings. The only exception is the BLEU performance of LiEGe compared to BART in the comprehensive setting, when the two models are only trained on the MIMICS-CEG dataset. A possible reason for this observation is that MIMICS data is rather small, and thus not enough for an effective training of the additional parameters in LiEGe compared to BART. This impact is also evident when the performance of BART (Wiki)/LiEGe (Wiki) is compared against its corresponding model BART/LiEGe, in both CEG and NEG settings. These comparisons show that knowledge learned from pre-training on a Wiki dataset can effectively transfer to real Web data of MIMICS.

To the best of our knowledge, NMIR [20] is the state-of-the-art model for generation of query intents/aspects. Using NMIR for aspect-oriented explanation, however, generates explanations with the least amount of diversity compared to other baselines. The main reason for this observation is that documents in the same cluster share the same explanation. This demonstrates that NMIR, and thus existing query intent generation models, do not address SeRE.

Table 7: Performance of ablated variants. Symbol $\triangledown$ shows statistical significant differences with LiEGe.

Dataset	Wiki-CEG		Wiki-NEG	
Metric	BLEU	R-L	BLEU	R-L
BART	16.53	42.23	11.84	31.84
LiEGe	18.79	44.32	**15.06**	**35.80**
LiEGe w/o MHP	18.81	44.36	$14.52^{\triangledown}$	$34.53^{\triangledown}$
LiEGe w/o OE	**18.87**	**44.39**	$14.23^{\triangledown}$	$34.58^{\triangledown}$
LiEGe w/o BA	$17.55^{\triangledown}$	$43.58^{\triangledown}$	$13.32^{\triangledown}$	$33.73^{\triangledown}$
LiEGe w/o CDA	$18.33^{\triangledown}$	$43.59^{\triangledown}$	$14.04^{\triangledown}$	$34.13^{\triangledown}$

Sample outputs. Table 6 shows the outputs of some models for two samples from the test set of Wiki-CEG and MIMICS-NEG. The document contents are not provided due to space limitations, and can be found in Wikipedia or the MIMICS dataset. In the example from Wiki-CEG, BART generates "history" and "culture", which are not wrong. However, the explanations from LiEGe are more detailed and informative in differentiating document contents. In the example from MIMICS-NEG, the aspect "side effects" is covered in d_1 and should not be repeated in the explanations for d_2 and d_3. HistGen and LiEGe perform better than other models in terms of not repeating previously generated explanations. LiEGe generates better explanations for this example compared to HistGen.

5.4 Ablation Study

We train and test variants of LiEGe, leaving one component out at a time, on Wiki-CEG and Wiki-NEG separately. The ablated versions are as follows.

LiEGe w/o OE does not add ordinal encodings to token embeddings in the encoder or the decoder.

LiEGe w/o MHP uses the [CLS] token embeddings in each layer as pooled document representations.

LiEGe w/o BA does not add contextualized document embeddings to the embeddings of their tokens. More specifically, Eq. (8) is

[4] https://en.wikipedia.org/wiki/Madrid

skipped and Eq. (9) becomes $E_t^{(h+1)} = \text{LN}(E_t^{(h)})$. A global layer thus only generates contextualized document embeddings as its output, and the final token embeddings from the encoder are not impacted by information from cross-document interactions.

LiEGe w/o CDA does not have cross-document attention sublayers in its decoding layers. In other words, this model incorporates cross-document interactions only during encoding.

The performance of ablated models are reported in Table 7. The results of models for generation of novelty explanations over Wiki-NEG show that LiEGe constantly outperforms its ablated versions and the observed improvements are statistically significant. Evaluation over Wiki-CEG for the comprehensive explanation however shows that LiEGe outperforms two of its ablated models where *broadcast & add* or *cross-document interactions* is removed. The performance differences with the other two ablated versions are not statistically significant. An on-par performance of LiEGe with the one without *ordinal document encoding* for comprehensive explanations is expected as these explanations are not dependent on the document position in a ranked list. Document representation by MHP is more important for novelty explanations compared to comprehensive ones. A possible reason for this observation is that the MHP representation of documents provides more flexibility to attend to a specific part of a document content compared to the [CLS] representation. This specific attention to a small part of a document is needed for novelty explanations, while comprehensive explanations can also be generated based on the [CLS] encoding of documents. Finally, *cross-document interactions* and *broadcast & add* are found to be essential for both novelty and comprehensive explanations. This demonstrates the necessity and utility of listwise modeling of the SeRE task.

6 CONCLUSION AND FUTURE WORK

We studied the problem of content-based explanation of search results in the two newly defined settings: novelty and comprehensive explanation generation. We proposed LiEGe that jointly explains all documents in a search results through exploiting cross-document interactions both in the encoder and the decoder. Experimental results demonstrate the effectiveness of LiEGe in explanation generation compared to state-of-the-art baselines. In the future, we would like to investigate the possibility of models for joint explanation generation and relevance ranking. To apply and adapt our explanation paradigm to search tasks other than ad-hoc information retrieval such as product search, is another interesting direction.

ACKNOWLEDGMENTS

This work was supported in part by the Center for Intelligent Information Retrieval and in part by NSF IIS-2039449. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsor.

REFERENCES

[1] Wasi Uddin Ahmad, Kai-Wei Chang, and Hongning Wang. 2019. Context attentive document ranking and query suggestion. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 385–394.

[2] Mohammad Aliannejadi, Hamed Zamani, Fabio Crestani, and W Bruce Croft. 2019. Asking clarifying questions in open-domain information-seeking conversations. In *Proceedings of the 42nd international acm sigir conference on research and development in information retrieval*. 475–484.

[3] Milad Alshomary, Nick Düsterhus, and Henning Wachsmuth. 2020. Extractive snippet generation for arguments. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1969–1972.

[4] Doug Beeferman and Adam Berger. 2000. Agglomerative clustering of a search engine query log. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*. 407–416.

[5] Sumit Bhatia, Debapriyo Majumdar, and Prasenjit Mitra. 2011. Query suggestions in the absence of query logs. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*. 795–804.

[6] Arbi Bouchoucha, Jing He, and Jian-Yun Nie. 2013. Diversified query expansion using conceptnet. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*. 1861–1864.

[7] Jaime Carbonell and Jade Goldstein. 1998. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*. 335–336.

[8] Ben Carterette and Praveen Chandar. [n.d.]. Probabilistic Models of Ranking Novel Documents for Faceted Topic Retrieval. In *CIKM'09*.

[9] Jun Chen, Xiaoming Zhang, Yu Wu, Zhao Yan, and Zhoujun Li. 2018. Keyphrase Generation with Correlation Constraints. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 4057–4066.

[10] Wanyu Chen, Fei Cai, Honghui Chen, and Maarten de Rijke. 2018. Attention-based hierarchical neural query suggestion. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 1093–1096.

[11] Wei-Fan Chen, Matthias Hagen, Benno Stein, and Martin Potthast. 2018. A user study on snippet generation: Text reuse vs. paraphrases. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 1033–1036.

[12] Wei-Fan Chen, Shahbaz Syed, Benno Stein, Matthias Hagen, and Martin Potthast. 2020. Abstractive Snippet Generation. In *Proceedings of The Web Conference 2020*. 1309–1319.

[13] Nachshon Cohen, Oren Kalinsky, Yftah Ziser, and Alessandro Moschitti. 2021. WikiSum: Coherent Summarization Dataset for Efficient Human-Evaluation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. 212–219.

[14] Van Dang and Bruce W Croft. 2010. Query reformulation using anchor text. In *Proceedings of the third ACM international conference on Web search and data mining*. 41–50.

[15] Zhicheng Dou, Sha Hu, Kun Chen, Ruihua Song, and Ji-Rong Wen. 2011. Multi-dimensional search result diversification. In *Proceedings of the fourth ACM international conference on Web search and data mining*. 475–484.

[16] Zi-Yi Dou, Pengfei Liu, Hiroaki Hayashi, Zhengbao Jiang, and Graham Neubig. 2021. GSum: A General Framework for Guided Neural Abstractive Summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 4830–4842.

[17] Maarten Grootendorst. 2020. KeyBERT: Minimal keyword extraction with BERT. https://doi.org/10.5281/zenodo.4461265

[18] Florian Haag, Qi Han, Markus John, and Thomas Ertl. 2014. Aspect Grid: A Visualization for Iteratively Refining Aspect-Based Queries on Document Collections.. In *GI-Jahrestagung*. 655–660.

[19] John A Hartigan and Manchek A Wong. 1979. Algorithm AS 136: A k-means clustering algorithm. *Journal of the royal statistical society. series c (applied statistics)* 28, 1 (1979), 100–108.

[20] Helia Hashemi, Hamed Zamani, and W Bruce Croft. 2021. Learning Multiple Intent Representations for Search Queries. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 669–679.

[21] Taher H Haveliwala. 2002. Topic-sensitive PageRank. In *Proceedings of the 11th international conference on World Wide Web*. 517–526.

[22] Jiyin He, Vera Hollink, and Arjen de Vries. 2012. Combining implicit and explicit topic representations for result diversification. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*. 851–860.

[23] Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. *Advances in neural information processing systems* 28 (2015), 1693–1701.

[24] Mayu Iwata, Tetsuya Sakai, Takehiro Yamamoto, Yu Chen, Yi Liu, Ji-Rong Wen, and Shojiro Nishio. 2012. Aspectiles: Tile-based visualization of diversified web search results. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*. 85–94.

[25] Bernard J Jansen, Amanda Spink, Judy Bateman, and Tefko Saracevic. 1998. Real life information retrieval: A study of user queries on the web. In *ACM Sigir Forum*, Vol. 32. ACM New York, NY, USA, 5–17.

[26] Zhengbao Jiang, Zhicheng Dou, and Ji-Rong Wen. 2016. Generating query facets using knowledge bases. *IEEE transactions on knowledge and data engineering* 29, 2 (2016), 315–329.

[27] Zhengbao Jiang, Ji-Rong Wen, Zhicheng Dou, Wayne Xin Zhao, Jian-Yun Nie, and Ming Yue. 2017. Learning to diversify search results via subtopic attention. In *Proceedings of the 40th international ACM SIGIR Conference on Research and Development in Information Retrieval*. 545–554.

[28] Rosie Jones, Benjamin Rey, Omid Madani, and Wiley Greiner. 2006. Generating query substitutions. In *Proceedings of the 15th international conference on World Wide Web*. 387–396.

[29] Weize Kong and James Allan. 2013. Extracting query facets from search results. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*. 93–102.

[30] Weize Kong and James Allan. 2014. Extending faceted search to the general web. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*. 839–848.

[31] Weize Kong and James Allan. 2016. Precision-oriented query facet extraction. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. 1433–1442.

[32] Reiner Kraft and Jason Zien. 2004. Mining anchor text for query refinement. In *Proceedings of the 13th international conference on World Wide Web*. 666–674.

[33] Victor Lavrenko and W Bruce Croft. 2001. Relevance based language models. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*. 120–127.

[34] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 7871–7880.

[35] Chenliang Li, Weiran Xu, Si Li, and Sheng Gao. 2018. Guiding generation for abstractive text summarization based on key information guide network. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. 55–60.

[36] Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*. 74–81.

[37] Hui Liu, Qingyu Yin, and William Yang Wang. 2019. Towards Explainable NLP: A Generative Explanation Framework for Text Classification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 5570–5581.

[38] Jiongnan Liu, Zhicheng Dou, Xiaojie Wang, Shuqi Lu, and Ji-Rong Wen. 2020. DVGAN: A Minimax Game for Search Result Diversification Combining Explicit and Implicit Features. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 479–488.

[39] Yang Liu and Mirella Lapata. 2019. Hierarchical Transformers for Multi-Document Summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 5070–5081.

[40] Ilya Loshchilov and Frank Hutter. 2018. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*.

[41] Sean MacAvaney, Craig Macdonald, Roderick Murray-Smith, and Iadh Ounis. 2021. IntenT5: Search Result Diversification using Causal Language Models. *arXiv preprint arXiv:2108.04026* (2021).

[42] Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*. 404–411.

[43] Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gulçehre, and Bing Xiang. 2016. Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*. 280–290.

[44] Shashi Narayan, Shay B Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. *arXiv preprint arXiv:1808.08745* (2018).

[45] Preksha Nema, Mitesh M Khapra, Anirban Laha, and Balaraman Ravindran. 2017. Diversity driven attention model for query-based abstractive summarization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 1063–1072.

[46] Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage Re-ranking with BERT. *arXiv preprint arXiv:1901.04085* (2019).

[47] Liang Pang, Jun Xu, Qingyao Ai, Yanyan Lan, Xueqi Cheng, and Jirong Wen. 2020. Setrank: Learning a permutation-invariant ranking model for information retrieval. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 499–508.

[48] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*. 311–318.

[49] Rama Kumar Pasumarthi, Honglei Zhuang, Xuanhui Wang, Michael Bendersky, and Marc Najork. 2020. Permutation equivariant document interaction network for neural learning to rank. In *Proceedings of the 2020 ACM SIGIR on International Conference on Theory of Information Retrieval*. 145–148.

[50] Ramakanth Pasunuru, Asli Celikyilmaz, Michel Galley, Chenyan Xiong, Yizhe Zhang, Mohit Bansal, and Jianfeng Gao. 2021. Data Augmentation for Abstractive Query-Focused Multi-Document Summarization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 13666–13674.

[51] Matt Post. 2018. A Call for Clarity in Reporting BLEU Scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*. 186–191.

[52] Shrimai Prabhumoye, Kazuma Hashimoto, Yingbo Zhou, Alan W Black, and Ruslan Salakhutdinov. 2021. Focused Attention Improves Document-Grounded Generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 4274–4287.

[53] Xubo Qin, Zhicheng Dou, and Ji-Rong Wen. 2020. Diversifying Search Results using Self-Attention Network. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 1265–1274.

[54] Filip Radlinski, Martin Szummer, and Nick Craswell. 2010. Inferring query intent from reformulations and clicks. In *Proceedings of the 19th international conference on World wide web*. 1171–1172.

[55] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research* 21, 140 (2020), 1–67.

[56] Razieh Rahimi, Youngwoo Kim, Hamed Zamani, and James Allan. 2021. Explaining Documents' Relevance to Search Queries. *arXiv preprint arXiv:2111.01314* (2021).

[57] Daniël Rennings, Felipe Moraes, and Claudia Hauff. 2019. An axiomatic approach to diagnosing neural IR models. In *European Conference on Information Retrieval*. Springer, 489–503.

[58] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. Model-agnostic interpretability of machine learning. *arXiv preprint arXiv:1606.05386* (2016).

[59] Rodrygo L. T. Santos, Craig Macdonald, and Iadh Ounis. 2015. Search Result Diversification. *Found. Trends Inf. Retr.* 9, 1 (March 2015), 1–90.

[60] Abigail See, Peter J Liu, and Christopher D Manning. 2017. 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)*. 1073–1083.

[61] Jaspreet Singh and Avishek Anand. 2019. Exs: Explainable search using local model agnostic interpretability. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*. 770–773.

[62] Jaspreet Singh and Avishek Anand. 2020. Model agnostic interpretability of rankers via intent modelling. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*. 618–628.

[63] Jaspreet Singh, Megha Khosla, Wang Zhenye, and Avishek Anand. 2021. Extracting per Query Valid Explanations for Blackbox Learning-to-Rank Models. In *Proceedings of the 2021 ACM SIGIR International Conference on Theory of Information Retrieval*. 203–210.

[64] Zhan Su, Zhicheng Dou, Yutao Zhu, Xubo Qin, and Ji-Rong Wen. 2021. *Modeling Intent Graph for Search Result Diversification*. ACM, 736–746.

[65] Jiwei Tan, Xiaojun Wan, and Jianguo Xiao. 2017. Abstractive document summarization with a graph-based attentional neural model. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 1171–1181.

[66] Paul Thomas, Bodo Billerbeck, Nick Craswell, and Ryen W. White. 2019. Investigating Searchers' Mental Models to Inform Search Explanations. 38, 1, Article 10 (Dec. 2019), 25 pages.

[67] TREC. 2000. Text REtrieval Conference (TREC) Data - English Relevance Judgements. https://trec.nist.gov/data/reljudge_eng.html.

[68] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.

[69] Manisha Verma and Debasis Ganguly. 2019. LIRME: locally interpretable ranking model explanation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1281–1284.

[70] Michael Völske, Alexander Bondarenko, Maik Fröbe, Benno Stein, Jaspreet Singh, Matthias Hagen, and Avishek Anand. 2021. Towards Axiomatic Explanations for Neural Ranking Models. In *Proceedings of the 2021 ACM SIGIR International Conference on Theory of Information Retrieval*. 13–22.

[71] Qinglei Wang, Yanan Qian, Ruihua Song, Zhicheng Dou, Fan Zhang, Tetsuya Sakai, and Qinghua Zheng. 2013. Mining subtopics from text fragments for a web query. *Information retrieval* 16, 4 (2013), 484–503.

[72] Xuanhui Wang and ChengXiang Zhai. 2008. Mining term association patterns from search logs for effective query reformulation. In *Proceedings of the 17th ACM conference on Information and knowledge management*. 479–488.

[73] Yumo Xu and Mirella Lapata. 2020. Coarse-to-fine query focused multi-document summarization. In *Proceedings of the 2020 Conference on empirical methods in natural language processing (EMNLP)*. 3632–3645.

[74] Le Yan, Zhen Qin, Rama Kumar Pasumarthi, Xuanhui Wang, and Michael Bendersky. 2021. Diversification-Aware Learning to Rank using Distributed Representation. In *Proceedings of the Web Conference 2021*. 127–136.

[75] Jin-ge Yao, Xiaojun Wan, and Jianguo Xiao. 2017. Recent advances in document summarization. *Knowledge and Information Systems* 53, 2 (2017), 297–336.

[76] Puxuan Yu, Razieh Rahimi, Zhiqi Huang, and James Allan. 2020. Learning to Rank Entities for Set Expansion from Unstructured Data. In *Proceedings of the 2020 ACM SIGIR on International Conference on Theory of Information Retrieval*. 21–28.

[77] Hamed Zamani, Susan Dumais, Nick Craswell, Paul Bennett, and Gord Lueck. 2020. Generating clarifying questions for information retrieval. In *Proceedings of The Web Conference 2020*. 418–428.

[78] Hamed Zamani, Gord Lueck, Everest Chen, Rodolfo Quispe, Flint Luu, and Nick Craswell. 2020. Mimics: A large-scale data collection for search clarification. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 3189–3196.

[79] Hamed Zamani, Bhaskar Mitra, Everest Chen, Gord Lueck, Fernando Diaz, Paul N Bennett, Nick Craswell, and Susan T Dumais. 2020. Analyzing and learning from user interactions for search clarification. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1181–1190.

[80] Ruqing Zhang, Jiafeng Guo, Yixing Fan, Yanyan Lan, and Xueqi Cheng. 2019. Outline generation: Understanding the inherent content structure of documents. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 745–754.

[81] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. BERTScore: Evaluating Text Generation with BERT. In *International Conference on Learning Representations*.

[82] Honglei Zhuang, Xuanhui Wang, Michael Bendersky, Alexander Grushetsky, Yonghui Wu, Petr Mitrichev, Ethan Sterling, Nathan Bell, Walker Ravina, and Hai Qian. 2021. Interpretable Ranking with Generalized Additive Models. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining* (Virtual Event, Israel) *(WSDM '21)*. 499–507.