Embedding Knowledge for Document Summarization: A Survey

Yutong Qu\textsuperscript{1}, Wei Emma Zhang\textsuperscript{1}, Jian Yang\textsuperscript{2}, Lingfei Wu\textsuperscript{3}, Jia Wu\textsuperscript{2} and Xindong Wu\textsuperscript{4}

\textsuperscript{1}School of Computer Science, The University of Adelaide, Australia
\textsuperscript{2}School of Computing, Macquarie University, Australia
\textsuperscript{3}JD.COM Silicon Valley Research Center, USA
\textsuperscript{4}Key Laboratory of Knowledge Engineering with Big Data, Hefei University of Technology, China
{yutong.qu,wei.e.zhang}@adelaide.edu.au, \{jian.yang, jia.wu\}@mq.edu.au, lwu@email.wm.edu, xwu@hfut.edu.cn

Abstract

Knowledge-aware methods have boosted a range of Natural Language Processing applications over the last decades. With the gathered momentum, knowledge recently has been pumped into enormous attention in document summarization research. Previous works proved that knowledge-embedded document summarizers excel at generating superior digests, especially in terms of informativeness, coherence, and fact consistency. This paper pursues to present the first systematic survey for the state-of-the-art methodologies that embed knowledge into document summarizers. Particularly, we propose novel taxonomies to recapitulate knowledge and knowledge embeddings under the document summarization view. We further explore how embeddings are generated in learning architectures of document summarization models, especially in deep learning models. At last, we discuss the challenges of this topic and future directions.

1 Introduction

With the exponential burst of textual data, demands in condensing voluminous text contents have been ubiquitous, bringing document summarization one of the most immensely researched fields in Natural Language Processing (NLP). Document Summarization (DS) aims to generate an abridged version of single (for Single DS, SDS) or multiple (for Multiple DS, MDS) topic-related texts as concise and coherent as possible while preserving the salient and factually consistent information [Ma \textit{et al.}, 2020]. There are two general methods in document summarization: 1) Extractive Document Summarization (EDS) respects the lexicon of the original text, regarding the summary formation is verbatim by key words and phrases selected from the source corpus; and 2) Abstractive Document Summarization (ADS) respects the semantics of the original text, regarding the summary construction is by rephrasing texts from the comprehension of text substances.

Generally, a DS model is to achieve the following goals [El-Kassas \textit{et al.}, 2021]:

- **G1. Coverage** of generating a summary that covers all the main and noteworthy contents of the input text(s);
- **G2. Non-redundancy** of generating a summary without any redundant or meaninglessly repeated information;
- **G3. Readability** of generating a summary composed by human-readable and coherent sentences to the viewer;
- **G4. Relevancy** of identifying related information within multiple input texts while generating the summary.

For MDS, an additional goal is [Ferreira \textit{et al.}, 2014]:
- **G4. Relevancy** of identifying related information within multiple input texts while generating the summary.

Recently, knowledge utilization in the summarization models has exhibited a huge potential for promoting the summarizer performance in terms of G1 to G4 and fuels one more DS capacity target:
- **G5. Factual Consistency** of generating a summary that obeys text facts and the commonsense of the real world.

In general, knowledge refers to the information acquired from facts and commonsense in source corpora and external sources [Hogan \textit{et al.}, 2021]. Empirical evidence from studies [Ji and Zhao, 2021; Tang \textit{et al.}, 2020; Wu \textit{et al.}, 2021; Chen \textit{et al.}, 2021] reported the worthwhile potentiality of fusing different kinds of knowledge in both extractive and abstractive document summarization methods for single or multiple inputs. Also of note are the possibility and motivated envisagement of effectively blending knowledge for DS to enrich the fact and commonsense consistency of the generated summaries. Throughout the knowledge usage in DS, from word-level knowledge [Han \textit{et al.}, 2016] to document-level knowledge [Yasunaga \textit{et al.}, 2017] and from internal knowledge [Tan \textit{et al.}, 2017] to external fact knowledge [Gunel \textit{et al.}, 2019], we observe that various formed knowledge appears and is incorporated into DS models in different ways. However, there is no existing work to summarize these researches.

| Surveys          | Coverage | Domain |
|------------------|----------|--------|
| Wang \textit{et al.}, 2017 | KGE      | NLP    |
| Cai \textit{et al.}, 2018 | KG; GE   | AI     |
| Xu, 2021         | KG; GE   | AI     |
| Ji \textit{et al.}, 2021 | KG; KGE  | NLP    |
| Hogan \textit{et al.}, 2021 | KG; KGE  | NLP    |
| Ours             | K; KE    | DS     |

Table 1: Outline of comparisons between existing relative surveys and ours. K, G, and E denote knowledge, graph, and embedding, respectively.
contributions. To fill this gap, we systematically investigate the knowledge and knowledge embedding methodologies under the DS view and report the results in this survey paper.

Comparisons to other surveys. Table 1 presents the comparisons between the most relevant literature review articles and this paper. These works focus on either graph embedding or knowledge graphs in a general manner. Neither of them targets a systematic view for one application. Differently, our survey studies the complete process of leveraging knowledge in a promising NLP application, document summarization: from acquiring knowledge to embedding knowledge, followed by how learning architectures generate and work with knowledge embeddings for summarizing documents. We select, describe and analyze the state-of-the-art works that embed knowledge into DS tasks and form the first systematic literature review of this kind.

Contributions of this survey. Our first contribution is a taxonomy of the knowledge leveraged in DS, presented in Section 2. In this paper, we consider all of the derived information in addition to the plain textual input as knowledge for DS, which is an expansion of the general factual knowledge. In our taxonomy, we broadly classify the knowledge in DS into four main categories: native knowledge, linguistic knowledge, semantic knowledge, and topical knowledge. The categorization is conducted lying on the layers of the knowledge from literalness to connotation implied in hierarchies of the documentary information from word to full text. The subcategories are also discussed. Knowledge embeddings refer to low-dimensional and continuous representations of knowledge [Wang et al., 2017], profiting better ways to permit various discrete-formed knowledge to be incorporated into learning models. Due to different knowledge leveraged, a wide variety of knowledge embedding methodologies have been employed in DS. Our second contribution is introducing a taxonomy of the existing knowledge embedding methodologies in DS tasks in Section 3 and how DS learning architectures generate different knowledge embeddings in Section 4. Consequently, we provide our envision about the future directions on the unfilled gaps and existing issues that are aligned with the goals G1 to G5 in Section 5, which forms our third contribution, in front of the survey conclusion in Section 6.

2 Knowledge Taxonomy

In this paper, we classify the group of knowledge that incorporated in document summarization models into four main categories. The knowledge is obtained from the literal text or latent semantic space and can work alone or by merging to derive high-level information for the goals G1 to G5.

2.1 Native knowledge

Native knowledge is the raw and plain textual data in the source text garnered without any filtration or transformation, such as origin words and sentences, typically leveraged as the auxiliary information [Zhang et al., 2020]. This knowledge is in the form of non-graph structures, such as token vectors, directly embedded into the model. It can represent the maximal amount of the original text information and enhance the content richness for promoting to achieve the goal G1.

2.2 Linguistic knowledge

Linguistic knowledge focuses on the source text information, such as the lexis, syntax, and grammar, presented as gauged word relations or parsed dependency relations.

Lexical knowledge. It is the estimated lexical relation knowledge among entities, such as centrality [Erkan and Radev, 2004], textual similarity [Yasunaga et al., 2017; Fan et al., 2019; Li et al., 2020; Wang et al., 2020; Ji and Zhao, 2021; Zhou et al., 2021; Chen et al., 2021], semantic similarity [Han et al., 2016; Tan et al., 2017; Zhang et al., 2020], and salience [Yasunaga et al., 2017; Tan et al., 2017]. This knowledge is in the form of numerical scores, infused as weights in learning models. It concedes to filter the relevant and salient text units for generating informative and succinct summaries, tallying with the goals G1 and G2. Also, its captured word relations can enhance summary coherence for the goal G3.

Syntactic knowledge. It involves syntactic dependency relations extracted by dependency parsers, such as the JAMR [Flanigan et al., 2014], CoreNLP dependency parser [Hermann et al., 2015], and neural dependency parser [Dozat and Manning, 2017]. This dependency relation forms the syntactic knowledge among words in a sentence, commonly modelled as dependency trees. Its preserved syntactic relations can assist determine redundant units and improve the summary coherence forward the goals G2 and G3.

Discourse knowledge. It covers discourse dependency relations concluded by discourse relation indicators via discovering deverbual noun references, event/entity continuations, discourse markers, or coreferent mentions [Yasunaga et al., 2017; Li et al., 2020; Xu et al., 2020], or by discourse parsers [Ji and Eisenstein, 2014]. This knowledge is in the form of a real number or a triplet (subject, predicate, object), commonly modelled as discourse graphs. It contains both syntactic and semantic information, excelling redundancy recognition and logic enhancement, profiting the goals G2 and G3.

2.3 Semantic knowledge

Semantic knowledge concentrates on concepts and facts gathered from the real world or extracted from the source text, typically preserved in the knowledge graph (KG).

Closed knowledge. It is the lexical relationship knowledge from the existing open-source and graph-based databases that contain general commonsense and human knowledge, such as WordNet [Miller, 1995], FrameNet [Ruppenhofer et al., 2006], ConceptNet5 [Speer and Havasi, 2012], and Wikidata [Vrandečić and Krötzsch, 2014]. This knowledge is in the form of the triplet (subject, predicate, object), preserved in KGs. Its involved real-world facts are surpassed to detect inconsistent fact errors in DS for achieving the goal G5.

Open knowledge. It is the ever-evolving and expansible lexis knowledge of semantic relations extracted and accumulated from the source corpora by open information extractors, such Open-domain Information Extraction (OpenIE) models [Angelii et al., 2015; Stanovsky et al., 2018]. This knowledge is in the form of the triplet (subject, predicate, object), modelled as KGs. The semantic relations it served can help improve summary concision and logic, promoting the goals G2 and G3.
2.4 Topical knowledge

Topical knowledge is the latent knowledge of the source text, gained by topic models, such as the Latent Dirichlet Allocation (LDA) [Blei et al., 2003] or neural topic model (NTM) [Miao et al., 2017]. This knowledge comprises topic salience [Zheng et al., 2019] and topical relevance [Cui et al., 2020; Li et al., 2020]. It can indicate the phrase-level semantic information to enhance summary coherence for the goal G2 or document-level semantic information for capturing relations among documents, benefiting the goal G3.

3 Knowledge Embedding Taxonomy

Native knowledge is usually in the original textual form, and its embedding relies on the embedding of the textual components in the document, such as token embedding, word embedding, sentence embedding and document embedding. Linguistic knowledge could be formed as texts or relations, and the latter is commonly modelled as a graph. Therefore, embedding linguistic knowledge covers both text embedding and graph embedding. Semantic knowledge similarly leverages both textual embedding and graph embedding. Topical knowledge is usually in data distribution form and requires to embed the distributions. In order to present the knowledge embedding applied in DS clearly, instead of grouping the embedding methods according to knowledge categorization, we propose a new taxonomy for knowledge embedding methods, as shown in Figure 1.

Text Embedding. Many knowledge embedding methods in DS focus on using textual contents from the source corpus.

1. Token embedding [Gunal et al., 2019; Liu and Lapata, 2019; Ji et al., 2020; Yuan et al., 2020; Huang et al., 2020; Wang et al., 2020; Liu et al., 2021; Ji and Zhao, 2021; Pasunuru et al., 2021; Zhou et al., 2021] which is generally produced from input tokens by the last layer of the language model (e.g., BERT). The WordPiece embedding [Zhang et al., 2019a] is a special token embedding obtained by WordPiece tokenizers.

2. Word embedding [Han et al., 2016; Takase et al., 2016; Tan et al., 2017; Zhang et al., 2017; Guan et al., 2019; Koncel-Kedziorski et al., 2019; Fan et al., 2019; Zheng et al., 2019; Zhang et al., 2019a; Ji et al., 2020; Jin et al., 2020; You et al., 2021] which is typically denoted as a vector of low dimension real numbers via methods, such as the one-hot vector and distributed representation. The Word2Vec is a general word embedding algorithm, producing the Word2Vec embedding [Yasunaga et al., 2017; Anh and Trang, 2019]. Also, the word vector garnered by the Global Vectors ForWord Representation (GloVe) algorithm and the FastText mechanism is known as the GloVe embedding [Zhang et al., 2017; Tan et al., 2017; Wang et al., 2020; Ji and Zhao, 2021] and the FastText embedding [Anh and Trang, 2019], respectively. Moreover, the Context embedding [Takase et al., 2016; Liu and Lapata, 2019] is a contextual vector for output words from the top layer of the language model (e.g., BERT), mapped with a weight matrix.

3. Sentence embedding [Tan et al., 2017; Zheng et al., 2019; Tang et al., 2020] which is typically a concatenation of word embeddings or gained by Sent2Vec [Zhang et al., 2017]. In deep neural summarization methods, sentence embeddings are computed from word embeddings [Yasunaga et al., 2017] or derived by language models (e.g., BERT). Besides, the TF-IDF is a general sentence embedding algorithm for the TF-IDF embedding [Tang et al., 2020]. The term frequency value is neglected in case the summary is formed by tremendously fewer tokens than the source document, known as the IDF-weighted word embedding [You et al., 2021]. Also, an Elementary Discourse Unit (EDU) is a sub-sentence phrase unit originating from RST discourse trees, represented by the EDU embedding [Xu et al., 2020]. The Phrase embedding [Koncel-Kedziorski et al., 2019] is a special sentence embedding produced from word embeddings run over last hidden states of neural networks (e.g., RNN). The Title embedding [Koncel-Kedziorski et al., 2019] is the title word embedding, regarding the title as a sentence, produced by neural networks (e.g., RNN) with last hidden states.

4. Document embedding [Yasunaga et al., 2017; Zheng et al., 2019; Zhang et al., 2019a] which is the concatenation of sentence embeddings or computed from sentence embeddings by the neural model (e.g., RNN and BERT).

5. Cluster embedding [Yasunaga et al., 2017] which is resulted from averaging document embeddings, supplied in the form of real numbers.

Graph Embedding. Graph embedding methods can be applied on embedding different components of the graph.

1. Node embedding [Liu et al., 2015; Takase et al., 2016;
Jin et al., 2020; Liu et al., 2021; Zhu et al., 2021] which represents a graph node, computed by the network layer from aggregated local graph information of its adjacent nodes and relations. In terms of the node orientation, the node embedding can be further classified into Forward-looking node embedding and Backward-looking node embedding [Koncel-Kedziorski et al., 2019].

2. **Entity embedding** [Gunel et al., 2019; Zhou et al., 2021] which is a representation of a graph entity, learned from output vectors of language models (e.g., BERT) or by techniques for modelling multi-relational data, such as the TransE [Bordes et al., 2013].

3. **Edge embedding** [Takase et al., 2016] which is the representation of an out-edge of the graph directed to the local parent node or global root node.

4. **Relation embedding** [Liu et al., 2015; Gunel et al., 2019; Ji et al., 2020; Liu et al., 2021] which is the representation of a relationship or concept between entities, typically derived byTransE from the graph and known as the Concept embedding [Liu et al., 2021]. Besides, it can be captured by firstly aggregating node and edge embeddings and then transforming it via linear transformations followed by nonlinear activation functions (e.g., ReLU) [Jin et al., 2020]. Also, the relationship type can be indicated by the Relation-type embedding [Ji et al., 2020].

5. **Graph weight embedding** [Fan et al., 2019] which is capable to represent the weight of both the node and edge of a graph, learned from the gating function or discretization of real numbers. A graph weight embedding that solely indicates the edge weight is represented as a token embedding and is equal to the number of merge operations increased by one, known as the Edge weight embedding [Wang et al., 2020; Zhou et al., 2021].

**Topic Embedding.** A Topic embedding [Zheng et al., 2019; Cui et al., 2020] is applied to embed topical information. It is a topic word vector typically composed from document embeddings or distilled subtopics. In deep learning architectures, it can be learned by neural topic models. Besides, the subtopic embedding [Zheng et al., 2019] is constructed by sentence embeddings.

**Position Embedding.** Position embedding is related to native knowledge. Its embedding is generated straightforward by using the token index information.

1. **Hard-position embedding** [Fan et al., 2019; Liu and Lapata, 2019; Yuan et al., 2020; Pasunuru et al., 2021; Wu et al., 2021] which is the numeric index of a token in its corresponding token sequence (i.e., sentence), also known as the Positional embedding [Zhang et al., 2020; Ji et al., 2020; Jin et al., 2020].

2. **Soft-position embedding** [Liu et al., 2021] which is the token index in a token sequence tree (i.e., sentence tree), represented as an integer number.

3. **Segment embedding** [Liu and Lapata, 2019; Yuan et al., 2020] which is a token notation assigned for discriminating multiple adjacent granularity levels (e.g., sentences) in a document, based on the parity of the level index.

## 4 Knowledge Embedding in Different Learning Architectures

Combining with the overview in previous sections, we discuss the DS knowledge embedding from the perspective of learning architectures applied for generating embeddings in DS models, with more attention to deep learning architectures. In addition, we broadly classify the reviewed DS research into four categories according to their adopted embedding approaches, as shown in Table 2. It covers main learning architectures for embeddings, types of DS tasks, usage of KGs, types of knowledge, main model architectures, and types of knowledge embeddings. Since techniques for obtaining knowledge embeddings have been almost thoroughly discussed in previous papers [Wang et al., 2017; Ji et al., 2021; Hogan et al., 2021], we herein only highlight specific technique characteristics deriving knowledge embeddings for DS.

### 4.1 RNN-based Approaches

The knowledge embedding leveraged in DS models with recurrent neural networks (RNN) [Koncel-Kedziorski et al., 2019; Zheng et al., 2019] are mostly combined with other word knowledge embeddings derived by Word2Vec, GloVe, or FastText. Besides, GraphWriter [Koncel-Kedziorski et al., 2019] employed the Science-domain Information Extraction (SciIE) for extracting science knowledge and embedded the knowledge by bidirectional RNNs in the Graph Transformer. Also, it is reported that RNNs can be combined with the graphical network in the DS model to effectively generate holistic cluster embeddings [Yasunaga et al., 2017]. In addition, long short-term memory network (LSTM) is broadly utilized for embedding knowledge with Word2Vec or FastText pre-trained embedding models [Anh and Trang, 2019], GloVe embedding models [Zhang et al., 2017], and BERT architectures [You et al., 2021]. Moreover, the tree-structured long short-term memory network (TreeLSTM) can be applied in DS to embed knowledge for representing the semantic knowledge mapped in AMR graphs [Takase et al., 2016].

### 4.2 Encoder-Decoder based Approaches

Transformers are initially utilized in DS models for capturing underlying contextual information in the input text [Vaswani et al., 2017]. With advances in Transformer architectures, Bidirectional and Auto-Regressive Transformers (BART), Bidirectional Encoder Representations from Transformers (BERT), and Robustly Optimized BERT Pretraining Approach (RoBERTa) has been adopted for learning knowledge embeddings in DS. In contrast to RoBERTa, BERT can embed knowledge tokenized by its sub-word tokenizer [Wu et al., 2021; Yuan et al., 2020], CoreNLP toolkit [Yuan et al., 2020; Zhang et al., 2019b], or WordPiece [Zhang et al., 2019a]. Also, it can combine with graph attention modules [Cui et al., 2020]. SemSUM [Jin et al., 2020] employed the Transformer encoder to learn knowledge embeddings from the syntactic knowledge extracted by an off-the-shelf dependency parser. BartGraphSumm [Huang et al., 2020] equipped the BART to encode the semantic knowledge extracted by the OpenIE from documents. EMSUM [Zhou et al., 2021] utilized the RoBERTa to embed the documentary information extracted by the Coreference Resolution Tool from AllenNLP.
Syntactic Knowledge (DS knowledge embedding approaches adopt the linearization representation, are applied in both abstractive and extractive word vector representation methods, such as the distributed approaches for DS [Guan et al., 2019]). Besides, advanced word vector representation methods, such as the distributed representation, are applied in both abstractive and extractive DS tasks, representing each word by its distributed representation [Han et al., 2016; Tan et al., 2017]. In addition, some DS knowledge embedding approaches adopt the linearization mechanism (e.g., TransE) to linearize the knowledge into sequences for embedding into the model architecture [Gan et al., 2019]. Scarcely approaches utilize straight the embedding algorithms, such as the TF-IDF algorithm, for learning knowledge embeddings in DS models [Tang et al., 2020].

**Table 2:** List of the representative Abstract or Extractive Single- or Multi-document summarization (DS) methods incorporated knowledge, indicating the usage of knowledge graphs (KG). Embedding kinds of TOK, WordPiece (WP), Word2Vec (W2V), GloVe (GV), FastText (FT), CONText, SENtence, TF-IDF, IDF, EDU, TIE, Document, CLUster, Node, ENTity, EDGe, Relation, Graph Weight (GW), Edge Weight (EW), TOPic, Position, and SEGment are presented with described Native Knowledge (NK), Lexical Knowledge (LK), Syntactic Knowledge (SK), Discourse Knowledge (DK), Closed Knowledge (CK), Open Knowledge (OK), and Topical Knowledge (TK).

4.3 GNN-based Approaches

The graph convolutional network (GCN) is a novel knowledge embedding approach in DS, majoring to embed graph-constructed knowledge, such as the knowledge graph ConceptNet5 [Ji et al., 2020; Xu et al., 2020]. As an upgrade from the GCN, the graph attention network (GAT) is widely utilized in DS for embedding knowledge extracted by the OpenIE or Stanford CoreNLP, preserved in graphs [Zhu et al., 2021; Ji and Zhao, 2021]. In addition, it is reported that embeddings learned by GATs can be combined with GloVe pre-trained embeddings [Wang et al., 2020; Ji and Zhao, 2021].

4.4 Non-deep learning approaches.

In contrast to DNN-based approaches, non-deep learning approaches are commonly adopted for embedding external knowledge into DS. The traditional word vector representation method is still utilized in recent knowledge embedding approaches for DS [Guan et al., 2019]. Besides, advanced word vector representation methods, such as the distributed representation, are applied in both abstractive and extractive DS tasks, representing each word by its distributed representation [Han et al., 2016; Tan et al., 2017]. In addition, some DS knowledge embedding approaches adopt the linearization mechanism (e.g., TransE) to linearize the knowledge into sequences for embedding into the model architecture [Gan et al., 2019].

5 Challenges and Future Opportunities

As still in its evolutionary stage, the research of embedding knowledge into document summarization faces numerous challenges and remains unfilled gaps. In this section, we discuss the challenges and promising avenues of ongoing and future works aligned with the goals G1 to G5.

5.1 Knowledge Quality

For DS models that leverage knowledge, the knowledge that covers less information, retains fault information, or contains factual errors can significantly harm the summarization performance. There are some latent future directions to maintaining the quality of a knowledge base for DS to carry large amounts of essential and factually consistent information.

Knowledge Collection. The issue of fact coverage can occur due to the choice of information extraction strategies when collecting knowledge from texts. This issue may cause losing prominent information from source documents in varying degrees, thus degrading the quality of the generated summary [Koncel-Kedziorski et al., 2019]. Therefore, an effective extraction strategy designed for improving the coverage in knowledge collection is requisite to be explored, in order to reduce the missing knowledge in the distilling process. It can also further help produce informative summaries corresponding to the goals G1 and G2. Besides, it is noted that determining the voice (i.e., active or passive) of sentences while extracting factual triples from the source text can advance the
extracted information quality [Abdi et al., 2017], which helps avoid missing information. More future research can be put in this direction to ensure the quality of collected knowledge.

**Knowledge Purification.** When operating the disambiguation process to the KG to eliminate redundancy knowledge for DS, some of the salient text information could be lost [Gunel et al., 2019]. Thus, a better strategy for KG disambiguation to condense the summary while remaining primary contents can be necessary. Also, more effective mechanisms for the entity recognition and linking of a KG to maintain relations of knowledge entities are worth investigating. These mechanisms can better reduce redundant information while retaining the knowledge base quality, achieving the goal G2.

**Knowledge Consistency.** Factual inconsistency errors refer to fact conflicts, categorized into contradicted fact (i.e., intrinsic error) and irrelevant fact (i.e., extrinsic error) to source text facts [Xie et al., 2021]. The knowledge from open-source KGs or extracted from the source corpora can inevitably involve varied intrinsic or extrinsic errors. Seriously erroneous knowledge can harm a knowledge-embedded summarizer’s performance terribly, mostly clashing with the goal G5. Even if factual inconsistency errors have been recognized and attached importance, scarce studies are qualified to precisely address and tackle inconsistency errors in DS. It is because inconsistency errors can be detected hardly by linguistic analysis. And since the knowledge databases for DS tasks are generally large-scale, it can cost laborious efforts to check each knowledge entity relation. Thus, exploring advanced ways to scan and solve inconsistent errors for incorporated external or personalized KGs, which contain plenty of mixed facts, can be a valuable future direction for knowledge-based DS.

### 5.2 Knowledge from Multi-facets

Except for investigations of manners to retain most of the superior knowledge, fusing knowledge from multiple facets to enhance the knowledge coverage in terms of different aspects can also promote the goals G1 to G5 for DS tasks.

**Knowledge from Multiple Resources.** Incorporating multiple types of knowledge from the real world and source corpora can be more beneficial in gathering more facts and prominent information in DS. It can improve the fact consistency of generated informative summaries, achieving the goals G1 and G5. As aforementioned in Section 2, recent DS studies reported the possibility and advantages of blending different kinds of knowledge, expanding the knowledge base and enhancing the commonsense uniformity. Also, integrating KGs with abstract meaning representations to combine knowledge for DS can be an appealing research direction [Liu et al., 2020]. However, empirically verifying the efficiency of leveraging fused knowledge and exploring effective multiple knowledge combinations are unattended research areas.

**Knowledge from Multiple Levels.** Recent works reported that sentence-level or paragraph-level extraction methods might lose global relationship information from the entire document context [Jain et al., 2020]. The lack of contextual relations can mainly reduce the summary coherence and harm the goal G3. Since most current extraction methods still focus on firstly splitting the entire document into sentences and then extracting triples from the sentence level, it inspires the research direction of leveraging the document-level extraction methodology. Moreover, employing the novel text-to-graph summarizer with the KG usage to capture more multi-level relations among knowledge can be a promising research direction in DS [Wu et al., 2020].

### 5.3 Knowledge Embedding Techniques

As summarized in Section 4, the majority of recent DS works utilized encoder-decoder models, neural networks, and non-deep learning knowledge embedding methods to generate embeddings when incorporating the knowledge into models. Explorations and experiments of employing novel knowledge embedding approaches, which widely benefit NLP tasks not limited to DS tasks, to achieve the goals G1 to G5 can be worthwhile future works.

**Novel KE Models.** The FocusE [Pai and Costabello, 2021] enhanced the knowledge preservation by merging number-encoded information with lexical knowledge triples while remaining the textual triple information. It served the ability to predict probability estimates of missing KG links. Besides, it presented the potential of combining multiple knowledge within one KG for further better knowledge embedding, improving the fact richness and accuracy for the goals G1 to G5. Since the ongoing extension work of FocusE is to associate multiple knowledge numerically to the same triple, future research on joining numbers with triples to conflate multifaceted knowledge within one graph for DS can be bright.

**Novel LM Models.** The K-BERT [Liu et al., 2020] can incorporate the KG into the Mask-Transformer architecture to embed knowledge, with token, soft-position, and segment embedding methods. And the CoLAKE [Sun et al., 2020] can jointly learn the word, token, soft-position, and type embeddings for KG entities and relations, enhancing the knowledge usage. The type embedding refers to a type notation of a graph node (i.e., word, entity, and relation), advancing to distinguishing nodes for the goals G2 and G3. Except for new knowledge embedding models, employing novel pre-training, deep learning, or reinforcement learning models [Sharma et al., 2019], which contain the potential forward the goals G1 to G5, can be a promising future direction for DS tasks.

### 6 Conclusion

Along with the pursuit of more informative and coherent summaries with factual consistency, attention to knowledge embedding as an incorporation module for document summarizers to enhance model performance and improve summary quality gathered pace. In this paper, we surveyed the state-of-the-art approaches to embedding knowledge into document summarization models. To explicitly review each representative knowledge embedding approach in document summarization, we proposed taxonomies for knowledge and knowledge embeddings and explored embedding learning architectures under the document summarization perspective. Furthermore, we discussed open questions and appealing research directions for embedding knowledge in document summarization tasks, which we hope can drive new improvements in the document summarization field.
References

[Abdi et al., 2017] A. Abdi, N. Idris, R. M. Alguliyev, and R. M. Alguliyev. Query-based multi-documents summarization using linguistic knowledge and content word expansion. *Soft Comput.*, 21:1785–1801, 2017.

[Angeli et al., 2015] G. Angeli, M. J. J. Premkumar, and C. D. Manning. Leveraging linguistic structure for open domain information extraction. In *ACL-IJCNLP*, 2015.

[Anh and Trang, 2019] D. T. Anh and N. T. Trang. Abstractive text summarization using pointer-generator networks with pre-trained word embedding. In *SoICT*, pages 473–478, 2019.

[Blei et al., 2003] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3, 2003.

[Bordes et al., 2013] A. Bordes, N. Usunier, A. Garcia-Durán, J. Weston, and O. Yakhnenko. Translating embeddings for modeling multi-relational data. In *NIPS*, 2013.

[Cai et al., 2018] H. Cai, V. Zheng, and K. Chang. A comprehensive survey of graph embedding: Problems, techniques, and applications. *IEEE TKDE*, 30, 2018.

[Chen et al., 2021] M. Chen, W. Li, J. Liu, X. Xiao, H. Wu, and H. Wang. SgSum: Transforming multi-document summarization into sub-graph selection. In *EMNLP*, 2021.

[Cui et al., 2020] P. Cui, L. Hu, and Y. Liu. Enhancing extractive text summarization with topic-aware graph neural networks. In *COLING*, pages 5360–5371, 2020.

[Dozat and Manning, 2017] T. Dozat and C. D. Manning. Deep biaffine attention for neural dependency parsing. In *ICLR*, 2017.

[El-Kassas et al., 2021] W. S. El-Kassas, C. R. Salama, A. A. Rafea, and H. K. Mohamed. Automatic text summarization: A comprehensive survey. *ESWA*, 165, 2021.

[Erkan and Radev, 2004] G. Erkan and D. R. Radev. LexRank: Graph-based lexical centrality as salience in text summarization. *J. Artif. Intell. Res.*, 22(1), 2004.

[Fan et al., 2019] A. Fan, C. Gardent, C. Braud, and A. Bordes. Using local knowledge graph construction to scale seq2seq models to multi-document inputs. In *EMNLP-IJCNLP*, pages 4186–4196, 2019.

[Ferreira et al., 2014] R. Ferreira, L. Cabral, F. Freitas, R. D. Lins, G. Silva, S. J. Simske, and L. Favaro. A multi-document summarization system based on statistics and linguistic treatment. *ESWA*, 41:5780–5787, 2014.

[Flanigan et al., 2014] J. Flanigan, S. Thomson, J. Carbonell, C. Dyer, and N. A. Smith. A discriminative graph-based parser for the abstract meaning representation. In *ACL*, pages 1426–1436, 2014.

[Guan et al., 2019] J. Guan, Y. Wang, and M. Huang. Story ending generation with incremental encoding and commonsense knowledge. In *AAAI-IAAI-EAAI*, 2019.

[Gunel et al., 2019] B. Gunel, C. Zhu, M. Zeng, and X. Huang. Mind the facts: Knowledge-boosted coherent abstractive text summarization. In *NIPS*, 2019.

[Han et al., 2016] X. Han, T. Lv, Z. Hu, X. Wang, and C. Wang. Text summarization using FrameNet-based semantic graph model. *Sci. Program.*, 2016:1–10, 2016.

[Hermann et al., 2015] K. M. Hermann, T. Kočiský, E. Gregor, T. Kociský, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom. Teaching machines to read and comprehend. In *NIPS*, pages 1693–1701, 2015.

[Hogan et al., 2021] A. Hogan, E. Blomqvist, M. Cochez, C. d’Amato, G. de Melo, C. Gutierrez, S. Kirrane, J. E. L. Gayo, R. Navigli, S. Neumaier, A. N. Ngomo, A. Polleres, S. M. Rashid, A. Rula, L. Schmelzeisen, J. Sequeda, S. Staab, and A. Zimmermann. Knowledge graphs. *ACM Comput. Surv.*, 54(4):1–37, 2021.

[Huang and Kurohashi, 2021] Y. J. Huang and S. Kurohashi. Extractive summarization considering discourse and coreference relations based on heterogeneous graph. In *EACL*, pages 3046–3052, 2021.

[Huang et al., 2020] L. Huang, L. Wu, and L. Wang. Knowledge graph-augmented abstractive summarization with semantic-driven cloze reward. In *ACL*, 2020.

[Jain et al., 2020] S. Jain, M. van Zuylen, H. Hajishirzi, and I. Beltagy. SciREX: A challenge dataset for document-level information extraction. In *ACL*, 2020.

[Ji and Eisenstein, 2014] Y. Ji and J. Eisenstein. Representation learning for text-level discourse parsing. In *ACL*, pages 13–24, 2014.

[Ji and Zhao, 2021] X. Ji and W. Zhao. SKGSUM: Abstractive document summarization with semantic knowledge graphs. In *IJCNN*, pages 1–8, 2021.

[Ji et al., 2020] H. Ji, P. Ke, S. Huang, F. Wei, X. Zhu, and M. Huang. Language generation with multi-hop reasoning on commonsense knowledge graph. In *EMNLP*, 2020.

[Ji et al., 2021] S. Ji, S. Pan, E. Cambria, P. Marttinen, and P. S. Yu. A survey on knowledge graphs: Representation, acquisition and applications. *IEEE TNNLS*, 2021.

[Jin et al., 2020] H. Jin, T. Wang, and X. Wan. SemSUM: Semantic dependency guided neural abstractive summarization. In *AAAI*, pages 8026–8033, 2020.

[Koncel-Kedziorski et al., 2019] R. Koncel-Kedziorski, D. Bekal, Y. Luan, M. Lapata, and H. Hajishirzi. Text generation from knowledge graphs with graph transformers. In *NAACL-HLT*, pages 2284–2293, 2019.

[Li et al., 2020] W. Li, X. Xiao, J. Liu, H. Wu, H. Wang, and J. Du. Leveraging graph to improve abstractive multi-document summarization. In *ACL*, 2020.

[Liu and Lapata, 2019] Y. Liu and M. Lapata. Text summarization with pretrained encoders. In *EMNLP-IJCNLP*, pages 3730–3740, 2019.

[Liu et al., 2015] F. Liu, J. Flanigan, S. Thomson, N. Sadeh, and N. A. Smith. Toward abstractive summarization using semantic representations. In *NAACL-HLT*, 2015.

[Liu et al., 2020] W. Liu, P. Zhou, Z. Zhao, Z. Wang, Q. Ju, H. Deng, and P. Wang. K-BERT: Enabling language representation with knowledge graph. In *AAAI*, 2020.
[Liu et al., 2021] Y. Liu, Y. Wan, L. He, H. Peng, and P. S. Yu. KG-BART: Knowledge graph-augmented BART for generative commonsense reasoning. In AAAI, pages 6418–6425, 2021.

[Ma et al., 2020] C. Ma, W. E. Zhang, M. Guo, H. Wang, and Q. Z. Sheng. Multi-document summarization via deep learning techniques: A survey. arXiv, abs/2011.04843, 2020.

[Miao et al., 2017] Y. Miao, E. Grefenstette, and P. Blunsom. Discovering discrete latent topics with neural variational inference. In ICMIL, pages 2410–2419, 2017.

[Miller, 1995] G. A. Miller. WordNet: A lexical database for English. Commun. ACM, 38(11):39–41, 1995.

[Pai and Costabello, 2021] S. Pai and L. Costabello. Learning embeddings from knowledge graphs with numeric edge attributes. In IJCAI, pages 2869–2875, 2021.

[Pasunuru et al., 2021] R. Pasunuru, M. Liu, M. Bansal, S. Ravi, and M. Dreyer. Efficiently summarizing text and graph encodings of multi-document clusters. In NAACL-HLT, pages 4768–4779, 2021.

[Ruppenhofer et al., 2006] J. Ruppenhofer, M. Ellsworth, M. R. L. Petruck, C. R. Johnson, and J. Scheffczyk. FrameNet II: Extended theory and practice. FrameNet Project, 2006.

[Sharma et al., 2019] E. Sharma, L. Huang, Z. Hu, and L. Wang. An entity-driven framework for abstractive summarization. In EMNLP-IJCNLP, pages 3280–3291, 2019.

[Speer and Havasi, 2012] R. Speer and C. Havasi. Representing general relational knowledge in ConceptNet 5. In LREC, pages 3679–3686, 2012.

[Stanovsky et al., 2018] G. Stanovsky, J. Michael, L. Zettlemoyer, and I. Dagan. Supervised open information extraction. In NAACL-HLT, pages 885–895, 2018.

[Sun et al., 2020] T. Sun, Y. Shao, X. Qiu, Q. Guo, Y. Hu, X. Huang, and Z. Zhang. CoLAKE: Contextualized language and knowledge embedding. In COLING, 2020.

[Takase et al., 2016] T. Takase, J. Suzuki, N. Okazaki, T. Hirao, and M. Nagata. Neural headline generation on abstract meaning representation. In EMNLP, 2016.

[Tan et al., 2017] J. Tan, X. Wan, and J. Xiao. Abstractive document summarization with a graph-based attentional neural model. In ACL, pages 1171–1181, 2017.

[Tang et al., 2020] T. Tang, T. Yuan, X. Tang, and D. Chen. Incorporating external knowledge into unsupervised graph model for document summarization. Electronics, 9, 2020.

[Vaswani et al., 2017] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In NIPS, 2017.

[Vrandečić and Krötzsch, 2014] D. Vrandečić and M. Krötzsch. Wikidata: A free collaborative knowledgebase. Commun. ACM, 57(10):78–85, 2014.

[Wang et al., 2017] Q. Wang, Z. Mao, B. Wang, and Li Guo. Knowledge graph embedding: A survey of approaches and applications. IEEE TKDE, 29(12):2724–2743, 2017.

[Wang et al., 2020] D. Wang, P. Liu, Y. Zheng, X. Qiu, and X. Huang. Heterogeneous graph neural networks for extractive document summarization. In ACL, pages 6209–6219, 2020.

[Wu et al., 2020] Z. Wu, R. Koncel-Kedziorski, M. Ostendorf, and H. Hajishirzi. Extracting summary knowledge graphs from long documents. arXiv, abs/2009.09162, 2020.

[Wu et al., 2021] W. Wu, W. Li, X. Xiao, J. Liu, Z. Cao, S. Li, H. Wu, and H. Wang. BASS: Boosting abstractive summarization with unified semantic graph. In ACL-IJCNLP, pages 6052–6067, 2021.

[Xie et al., 2021] Y. Xie, F. Sun, Y. Deng, Y. Li, and B. Ding. Factual consistency evaluation for text summarization via counterfactual estimation. In EMNLP, 2021.

[Xu et al., 2020] J. Xu, Z. Gan, Y. Cheng, and J. Liu. Discourse-aware neural extractive text summarization. In ACL, pages 5021–5031, 2020.

[Xu, 2021] M. Xu. Understanding graph embedding methods and their applications. SIREV, 63(4), 2021.

[Yasunaga et al., 2017] M. Yasunaga, R. Zhang, K. Meelu, A. Pareek, K. Srinivasan, and D. Radev. Graph-based neural multi-document summarization. In CoNLL, pages 452–462, 2017.

[You et al., 2021] J. You, C. Hu, H. Kamigaito, H. Takanura, and M. Okumura. Abstractive document summarization with word embedding reconstruction. In RANLP, pages 1586–1596, 2021.

[Yuan et al., 2020] R. Yuan, Z. Wang, and W. Li. Fact-level extractive summarization with hierarchical graph mask on BERT. In COLING, pages 5629–5639, 2020.

[Zhang et al., 2017] C. Zhang, S. Sah, T. Nguyen, D. K. Peri, A. C. Loui, C. Salvaggio, and R. W. Ptucha. Semantic sentence embeddings for paraphrasing and text summarization. IEEE GlobalSIP, pages 705–709, 2017.

[Zhang et al., 2019a] H. Zhang, J. Cai, J. Xu, and J. Wang. Pretraining-based natural language generation for text summarization. In CoNLL, pages 789–797, 2019.

[Zhang et al., 2019b] X. Zhang, F. Wei, and M. Zhou. HIBERT: Document level pre-training of hierarchical bidirectional transformers for document summarization. In ACL, pages 5059–5069, 2019.

[Zhang et al., 2020] J. Zhang, Y. Zhao, M. Saleh, and P. J. Liu. PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization. In ICML, 2020.

[Zheng et al., 2017] X. Zheng, A. Sun, J. Li, and K. Muthuswamy. Subtopic-driven multi-document summarization. In EMNLP-IJCNLP, 2019.

[Zhou et al., 2021] H. Zhou, W. Ren, G. Liu, B. Su, and W. Lu. Entity-aware abstractive multi-document summarization. In ACL-IJCNLP, pages 351–362, 2021.

[Zhu et al., 2021] C. Zhu, W. Hinthorn, R. Xu, Q. Zeng, M. Zeng, X. Huang, and M. Jiang. Enhancing factual consistency of abstractive summarization. In NAACL-HLT, pages 718–733, 2021.