SN Computer Science (2023) 4:500
https://doi.org/10.1007/s42979-023-01867-1

ORIGINAL RESEARCH

Graph-Enhanced Biomedical Abstractive Summarization Via Factual Evidence Extraction

Giacomo Frisoni[1] · Paolo Italiani[1] · Gianluca Moro[1] · Ilaria Bartolini[1] · Marco Antonio Boschetti[2] · Antonella Carbonaro[1]

Received: 5 November 2022 / Accepted: 3 April 2023

Abstract

Infusing structured semantic representations into language models is a rising research trend underpinning many natural language processing tasks that require understanding and reasoning capabilities. Decoupling factual non-ambiguous concept units from the lexical surface holds great potential in abstractive summarization, especially in the biomedical domain, where fact selection and rephrasing are made more difficult by specialized jargon and hard factuality constraints. Nevertheless, current graph-augmented contributions rely on extractive binary relations, failing to model real-world n-ary and nested biomedical interactions mentioned in the text. To alleviate this issue, we present EASumm, the first framework for biomedical abstractive summarization empowered by event extraction, namely graph-based representations of relevant medical evidence derived from the source scientific document. By relying on dual text-graph encoders, we prove the promising role of explicit event structures, achieving better or comparable performance than previous state-of-the-art models on the CDSR dataset. We conduct extensive ablation studies, including a wide experimentation of graph representation learning techniques. Finally, we offer some hints to guide future research in the field.

Keywords Abstractive document summarization · Event extraction · Semantic parsing · Biomedical text mining · Knowledge-driven natural language processing · Natural language understanding

Introduction

International experts argue that language is the highest manifestation of human intelligence [1]. This makes learning knowledge from text one of the greatest challenges of modern artificial intelligence. Language is ambiguous, consisting of several expressions that allude to the same fact and often lacking background knowledge for the mentioned entities. Despite capturing a massive amount of knowledge, current state-of-the-art language models—even with 10^{11} parameters—struggle to separate high-level semantics from language structure [2, 3], acting as memories rather than intelligent networks. Consequently, they notoriously suffer from hallucinations, biases, low robustness, and fragility (vulnerability to adversary attacks) [4, 5] that severely limit their real-world application. Semantics is central to summarization, where humans are asked to grasp the relevant parts of the input document, link them, and rephrase the selected entangled facts to create an original short text conveying as much information content as possible. These challenges are further emphasized by the biomedical literature, characterized by documents having domain-specific terminology, elaborated links among entities, no space for interpretation, and no tolerance for factual mistakes. However, automatic summarization systems can substantially help healthcare professionals have a quick and reasonably close overview of the knowledge encapsulated in large scientific corpora, outlining a prerogative toolbox for efficient knowledge discovery [6–9]. Standing on the shoulders of AI is even more important when we consider the accelerated speed of publication, which since 2020 has exceeded the threshold of 1 million new papers on PubMed per year (3 per minute) [10].

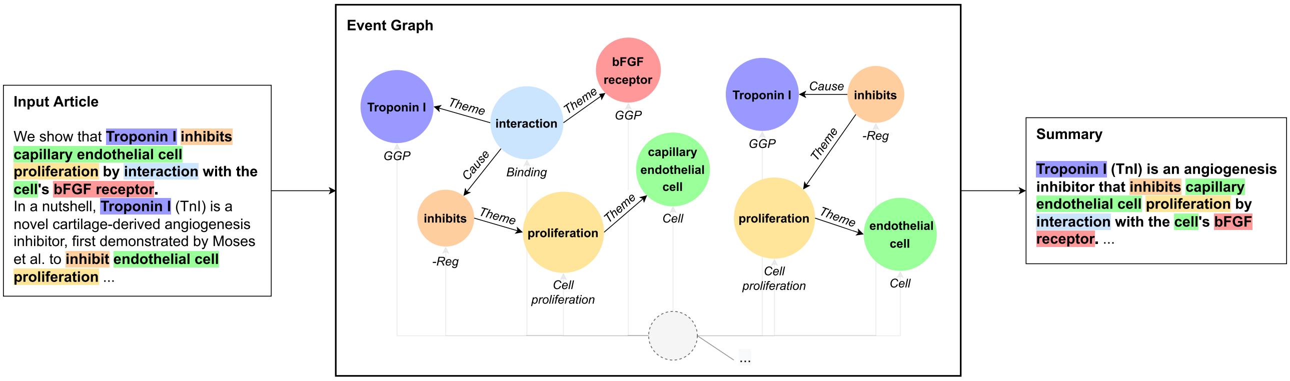
Researchers together have cast doubt about whether pure data-driven methods based on deep neural architectures

Giacomo Frisoni, Paolo Italiani and Gianluca Moro contributed equally to this work.

This article is part of the topical collection "Advances on Data Science, Technology and Applications" guest edited by Slimane Hammoudi, Alfredo Cuzzocrea and Oleg Gusikhin.

Extended author information available on the last page of the article

Published online: 29 June 2023



Input Article

We show that Troponin I inhibits capillary endothelial cell proliferation by interaction with the cell's bFGF receptor.
In a nutshell, Troponin I (TnI) is a novel cartilage-derived angiogenesis inhibitor, first demonstrated by Moses et al. to inhibit endothelial cell proliferation ...

Summary

Troponin I (TnI) is an angiogenesis inhibitor that inhibits capillary endothelial cell proliferation by interaction with the cell's bFGF receptor. ...

Fig. 1 Qualitative example of event-driven biomedical abstractive summarization. The event graph localizes relevant information for entities and triggers, providing a global context pivotal for generating a better-quality summary. Figure taken from [19]

would be sufficient to achieve understanding. The interpretability requirement and the lack of grounding to the core interactions expressed in the document beg the question of whether symbolic representations may come to the rescue. Following this intuition, the community has recently investigated the integration of structured knowledge into neural language models [11], reckoning on external knowledge graphs [12] or structured representations obtained via semantic parsing [13] and latent semantic correlations [14–16]. Depending on the graph's nature, neuro-symbolic methods may thus target amplified understanding or knowledge acquisition capabilities. Reaching the latter goal only with traditional pretraining approaches can be inefficient and expensive. For example, acquiring a fact like "Paracetamol can treat cold" requires having a large number of co-occurrences of "paracetamol" and "cold" concepts in the pretraining corpus. While the combination of language models and knowledge graphs is a research path already widely explored, the same does not apply to semantic graphs. Existing contributions merely extract flat, open-domain, and binary relations, which can result in inferring incomplete or incorrect facts non-useful for biomedicine [10, 17]. In this scenario, event extraction [10] appears as the most promising option for obtaining task-driven meaning representations. Under the umbrella of structured prediction, it aims to derive n-ary and potentially nested interactions between participants having a specific semantic role. We point out to the reader that events are released from the presence of a temporal element, and for clarity, we consider the following definition of event proposed in [10]:

> *"An event is a specific occurrence of something that happens and involves an arbitrary number of attributes and participants covering a specific semantic role, depending on the event type. The interaction (i.e., dynamic relation) modeled by an event represents or leads to some state change"*

In this paper, the keyword "event" therefore stays for medical evidence mentioned in the scientific literature, in accordance with previous works. Still, we are aware that this term may be misleading and requires revision [10].

We propose EASumm, the first model leveraging event extraction for abstractive single-document biomedical summarization, adopting a tandem architecture combining text and graph representations. We test our solution on the CDSR dataset [18], and we prove how biomedical event extraction contributes to reserving the essential global context and keeping the connection between the most relevant entities, thus generating a higher quality summary (see Fig. 1). Extensive ablation studies prove the contribution of each module and quantify the impact of multiple graph representation learning techniques.

This is an extension of [19], where we test multiple relation-aware graph representation learning modules, dissect event extraction in more details, clarify our methodology with details and algorithms, and openly release all code and data.

The rest of the paper is organized as follows. The following section examines related work. "Event extraction" provides a more extensive discussion on event extraction. Then, "Graph construction" presents our event-based strategy for deriving structured medical evidence from the source text. Next, "Model" details our model, from the architecture design to the training objectives. "Experimental setup" illustrates our experimental setup, while "Results" showcases the results obtained. Finally, "Conclusion" reports the conclusions and points out future directions.

Related Work

Abstractive Document Summarization

Summarizing text demands generating a concise summary discarding unnecessary attributes, and preserving the salient notions of the source document. Notably, abstractive summarization does not imply simply copying phrases from the source text but also coming up with new content, echoing a human-like interpretation and paraphrasis. Transformer-based language models have achieved astonishing results in recent years, mainly thanks to deep encoder-decoder architectures and self-supervised pretraining. In a nutshell, the encoder maps the source tokens into a sequence of continuous representations while the decoder reads them and autoregressively generates the summary one token at a time. Their ability to learn universal representations from large volumes of unlabeled text data and then transfer such knowledge to downstream tasks has revolutionized the abstractive summarization research sphere [20–25]—even in low-resource [26, 27] and multi-document settings [28]. Nevertheless, quantitative studies and large-scale human evaluations [29] have confirmed that current text generators are still heavily victims of hallucinations and prone to produce summaries that are unfaithful to the input documents. For this reason, the latest solutions are mostly knowledge-driven or tend to complement training with reinforcement learning modules to improve informativeness and consistency [30–32].

Graph-Enhanced Summarization

Human language is highly ambiguous, with multiple ways to express the same concept unit, where the underlying meaning is oftentimes altered by high-level linguistic constructs. Additionally, a single sentence may incorporate various predicate-argument structures. Despite these observations, current language models only consider the superficial organization of the text document, which is almost irrelevant to identifying its real and deeper semantic content [13]. Climbing towards natural language understanding, an increasing number of researchers argue that a model trained purely on the form will never learn the meaning, lacking signals to learn non-linguistic relations [33].

To this end, structured representations allow different quality improvements (e.g., coherence, factuality, low redundancy, long-range dependencies, informativeness, consistency) depending on how they are constructed. In particular, semantic parsing graphs normalize lexical and syntactic variations, providing formal meaning representations capable of decoupling concept units (*what to say*) from language competencies (*how to say it*).

Graph structures have long been used for extractive summarization. In this sense, early approaches, such as TextRank [34], propose unsupervised keyword and sentence extraction methods exploiting graph-based ranking algorithms to determine each vertex's importance. Extensions have been devised to incorporate document-level information [35] or introduce graph-based attention into encoder-decoder architectures [36]. As for abstractive summaries, results are mostly built on the cross-cutting success of graph neural networks (GNNs), a famed class of deep learning methods designed to process graph-represented data without imposing linearization or hierarchical constraints. Fernandes et al. [37] combine sequence encoders with GNNs feed with weakly-structured data inferred by the text through off-the-shell NLP tools, including named entity recognition and coreference resolution; the final model compares favorably with baselines using only the sequential or graphical structure. Structured summarization also relates to the graph verbalization trend [38–40], where inputs may originate from knowledge graphs, information extraction or semantic parsing techniques. Instead of tackling a graph-to-text approach, An et al. [41] redefines the task of scientific papers summarization by utilizing a graph-enhanced encoder on top of a citation network. To concretize a text-graph complementary view—where GNN channels are used in addition to traditional document encoding—many researchers have tried different ways of automatically building a machine-readable knowledge representation linked to the underlying text [42–44], also considering different level of granularities, like entities and sentences [45]. OpenIE [46] and Stanford CoreNLP [47] are undoubtedly the two most popular libraries, targeting triplets and coreference resolution, respectively.

Importantly, graph-LSTMs appear as one of the most effective ways for constructing graph-guided summarizers [32, 37, 39, 41, 44, 45], being competitive with large pretrained language models at a lower computational and environmental cost.

Event Extraction

Relation extraction (RE) systems primarily focus on highly-extractive binary relations, giving rise to a list of $< subject, predicate, object >$ triplets connecting only entity-mention pairs. Despite their simplicity, flat triplets in biomedical science are notoriously inadequate to capture the source document's complete biological meaning (see "Comparison with Relation Extraction"). Per contra, event extraction (EE) systems can handle *n*-ary complex relations with nested and overlapping definitions. Remarkably, the EE history is very intertwined with biomedicine.

According to the BioNLP-ST competitions [48–50], events are composed of a trigger (a text span which testifies their occurrence, e.g., "interacts", "regulates"), a type (e.g., "binding", "regularization"), and a set of arguments with a specific role (e.g., "cause", which can be typed entities or events themselves. Please note that the event schemas (i.e., target event, entity, and role types) are pre-established, conforming to a reference ontology. Hence, differently from linguistically grounded semantic parsing techniques like abstract meaning representation (AMR), EE is domain-specific and—given an input sentence—outputs a graph only in case of evidence of interest.

Comparison with Relation Extraction

EE and RE have a lot of common ground. They both aim to detect relations from raw text and build structured, machine-readable representations. In RE settings, a relation can be defined as $R = r(a_1, a_2, .., a_n)$ where r is a relation type and $a_i \ \forall i = 1, \dots n$ is typically an entity. When $n > 2$, we say that R is a complex relation. Most of the RE systems, such as Open-IE, are not capable of extracting complex relations, they usually detect general-domain directed or undirected binary relations ($n = 2$). The set of triples $r(a_1, a_2)$ might not be sufficient to represent underlying knowledge correctly, especially in biomedicine. This simplification of the original ground-truth complex structure may lead to the extraction of incomplete, uninformative, or erroneous facts [10, 17]. Events have been precisely designed to solve these limitations, targeting a set of sophisticated closed-domain interactions. Figure 2 illustrates a real-world biomedical example recapping the crucial expressiveness divergences between RE and EE outputs.

Biomedical Event Extraction

A series of datasets have been proposed to improve EE research. Among these, we highlight the series of BioNLP shared tasks (BioNLP-STs) [10]. The labeling process is curated by domain experts, resulting in gold standards that can be used for training or benchmarks. The availability and coverage of biomedical EE corpora are still retained by the extremely expensive annotating process. For instance, annotating the GENIA corpus—one of the most popular biomedical EE datasets —took 1.5 years with five part-time annotators and two coordinators [51]. Such complexity-motivated cost hinders the number of examples, with training sets that typically consist of < 300 instances. Another known problem is related to class imbalance, meaning that a large portion of event types might be under-represented.

Annotations for a certain text document (*.txt*) are saved in standoff *.a** files, where a distinction is made between *.a1* and *.a2*. Pointedly, an *.a1* file encodes information about gold entities; instead, an *.a2* file encodes information about triggers and the events rooted in them (i.e., reference multi-relational interconnections between entities and triggers). Figure 3 shows an example. Each line in an *.a** file refers to a single annotation. In turn, each annotation is made of multiple attributes separated by a single `TAB` character, always including an identifier. Entity and trigger annotations are accompanied by their type (e.g., "Gene" for an entity, "Localization" for an event), the (*start*, *end*) character offset of their mention, and the marked text. Instead, an event annotation consists of a `SPACE`-separated set comprising the trigger and the related arguments (entities or other triggers in case of sub-events). The event trigger is specified as `TYPE:ID`, thus identifying the event type and its trigger through the identifier. The event arguments are indicated as `ROLE:ID` pairs, thereby listing the semantic role and the argument identifier filling that role. So, by convention, the event type is stated both in the trigger and event annotation. Note that several events can share the same trigger and that, while the event trigger should be specified first, the event arguments can appear in any order.

Graph Construction

We construct graphs from raw documents applying DeepEventMine (shortened as DEM) [52], a sentence-level EE discriminative neural network with state-of-the-art results on seven biomedical tasks. DEM does not depend on gold entities but carries out named entity recognition in end-to-end without losing too much performance. Starting from SciBERT contextual representations [53], DEM enumerates all the possible text spans in a sentence up to a certain window length, then executes a joint detection and classification flow of (1) entities and triggers, (2) roles, (3) events and modifiers, through custom layers.

Following Frisoni et al. [54], we shape events as multi-relational graphs. Ergo, an event graph $G = (V, E)$ consists of a finite set of nodes $V = v_1, \dots v_V$—triggers or entities—and a set of edges $E \subseteq V \times V$ modeling entity-trigger or trigger–trigger relations, with the seconds applying for nested events. Edges are directed, labeled, and unweighted, with no cycles. Both nodes and edges in G are associated with type information; hence, the graph is heterogeneous and multi-relational. An edge $e_{i,j}$ connects node v_i to node v_j. Entities that don't belong to any event are ignored during graph construction. Node connections are encoded in an adjacency matrix $A \in \mathbb{R}^{V \times V}$, where $a_{ij} = 1$ if there is a directed link from v_i to v_j, and 0 otherwise. We operate graph rewiring by adding a master node connecting all event nodes to enhance

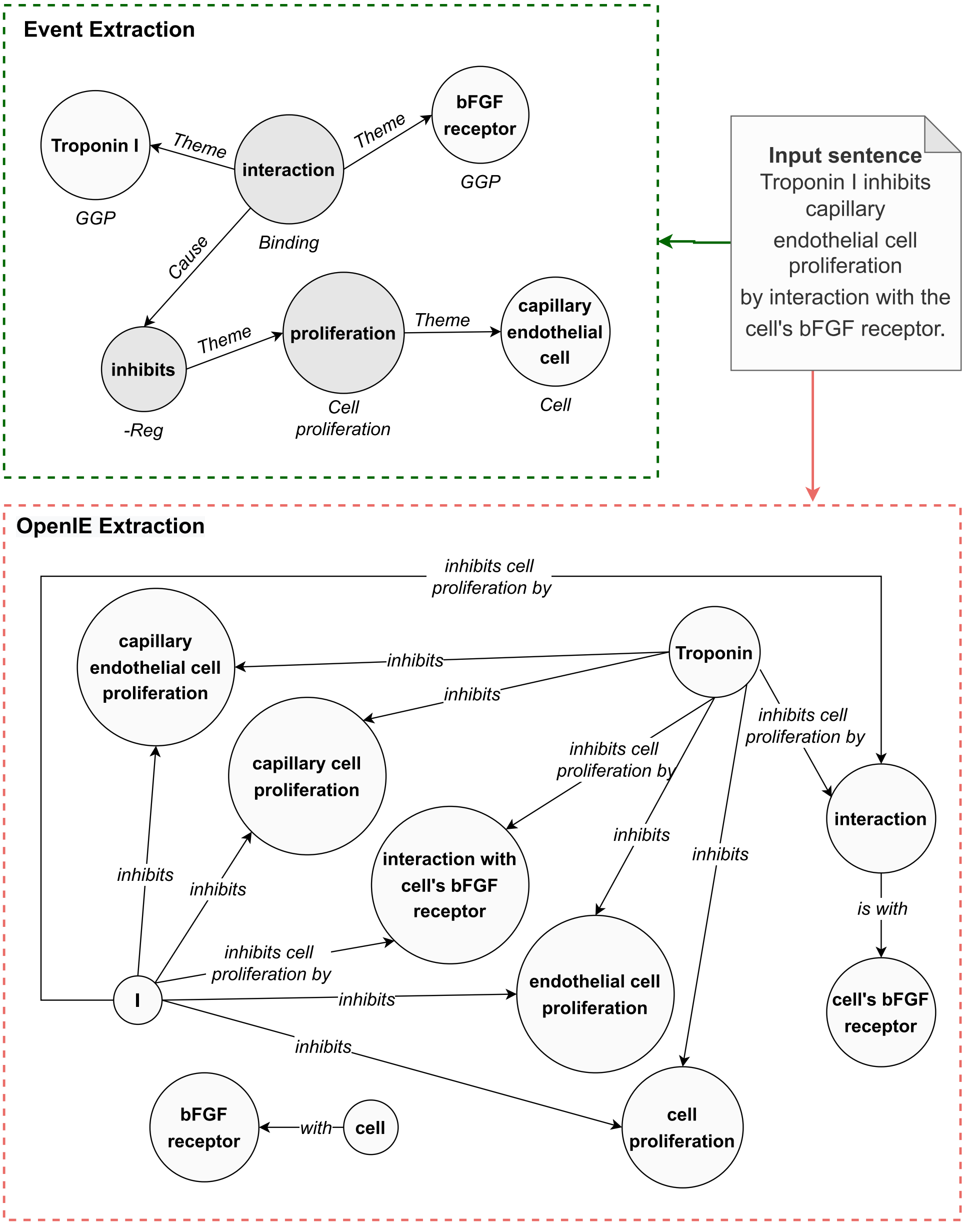
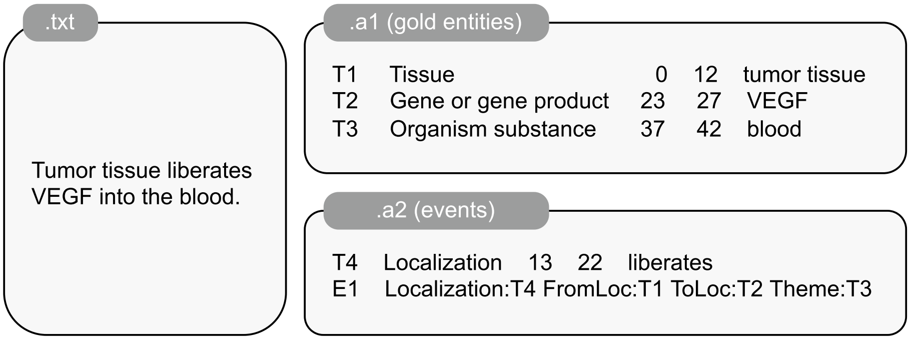

Event Extraction
Troponin I
GGP
Theme
interaction
Binding
Theme
bFGF receptor
GGP
Cause
inhibits
-Reg
Theme
proliferation
Cell proliferation
Theme
capillary endothelial cell
Cell
Input sentence
Troponin I inhibits capillary endothelial cell proliferation by interaction with the cell's bFGF receptor.
OpenIE Extraction
inhibits cell proliferation by
capillary endothelial cell proliferation
inhibits
Troponin
inhibits
capillary cell proliferation
inhibits cell proliferation by
inhibits cell proliferation by
interaction
interaction with cell's bFGF receptor
inhibits
inhibits
inhibits
inhibits
is with
inhibits cell proliferation by
I
inhibits
endothelial cell proliferation
cell's bFGF receptor
inhibits
bFGF receptor
with
cell
cell proliferation

Fig. 2 Comparison between semantic graphs obtained with closed-domain event extraction and open-domain relation extraction on a sentence taken from a PubMed article. The prediction enclosed in the green box comes from DeepEventMine MLEE, while the other is made with OpenIE 5.1 (https://github.com/dair-iitd/OpenIE-standalone). An event graph maps complex interactions mentioned in the text to a linkage between the trigger (dark gray) and entity (light gray) nodes, labeling edges and arguments with pre-defined roles and types aligned with an ontology. On the other hand, an OpenIE graph collects a possible set of triplets consisting of untyped text phrases. The OpenIE graph is merely extractive, error-prune, and devoid of additional metadata; worse, it does not capture semantic interconnections between n-ary participants, often ignoring crucial conditions for the correctness of a triplet or extracting incomplete facts difficult to merge with post-processing Figure taken from [19]



.txt

Tumor tissue liberates VEGF into the blood.

.a1 (gold entities)

```
T1   Tissue                   0    12   tumor tissue
T2   Gene or gene product     23   27   VEGF
T3   Organism substance       37   42   blood
```

.a2 (events)

```
T4   Localization   13   22   liberates
E1   Localization:T4 FromLoc:T1 ToLoc:T2 Theme:T3
```

Fig. 3 Example of.txt, .a1, and. a2 files

the information flow and ensure we end up with a single graph rather than a set of small disjoint graphs.

Model

Our model observes a biencoder-decoder architecture (depicted in Fig. 4), taking inspiration from [32]. It takes two inputs, the sequence of all tokens present in the document $x = x_k$ and the event graph G, constructed as explicated in “Graph construction”.

Document Encoders

The sequence of tokens x is fed to a bidirectional transformer-based encoder. We take token embeddings from the output of the last layer and pass them to a multi-layer bidirectional LSTM (BiLSTM), thus gaining the sequence of encoder hidden states h_k. We implemented the following BERT [55] variants.

SciBERT

SciBERT [53] performs pretraining on a multi-domain corpus of scientific publications containing 1.14 M biomedical and computer science papers. It uses an in-domain vocabulary (SciVocab), characterized by a 42% token overlap with respect to the original BERT vocabulary, spotlighting a substantial difference in frequently used words between scientific and general-domain texts.

RoBERTa

RoBERTa [56] provides an updated version of BERT by optimizing its training process. The model is pretrained longer, with bigger batches and over more data. The next sentence prediction objective is removed. Longer sequences are taken into account, and the masking pattern—applied to the training data—is dynamically changed.

Graph Encoders

Node Initialization

Each node feature v_i is initialized by taking into account both its text span and entity/trigger type. First, we average the per-token hidden states h_k corresponding to the matched text. Then, we concatenate the acquired representation to the argument type embedding s_a (or trigger type embedding s_t) learned by DEM. On this point, we believe that type metadata can play a vital role in augmenting the understanding capacity of the model and resolving ambiguities. The master node is represented by a 0-vector.

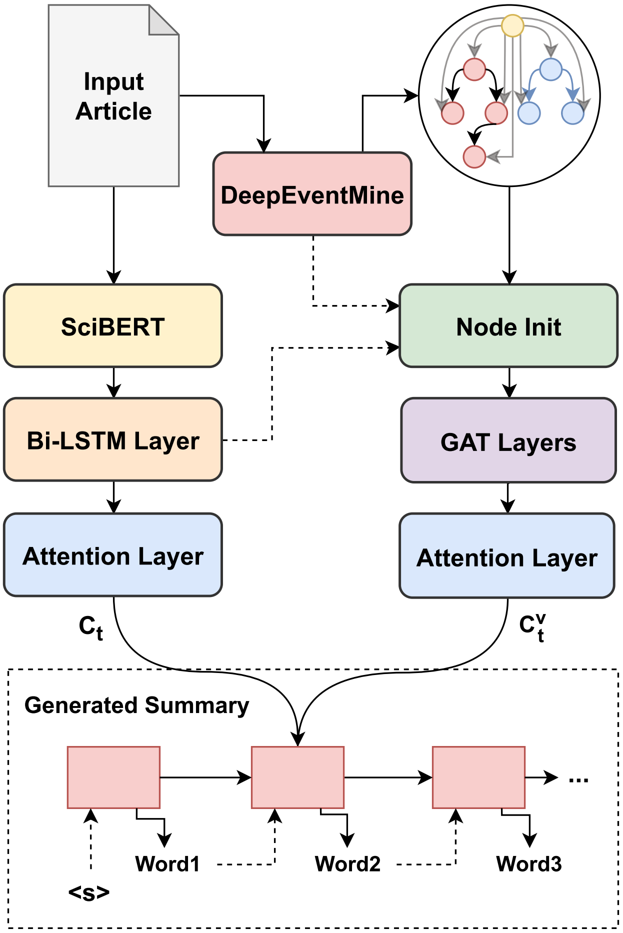
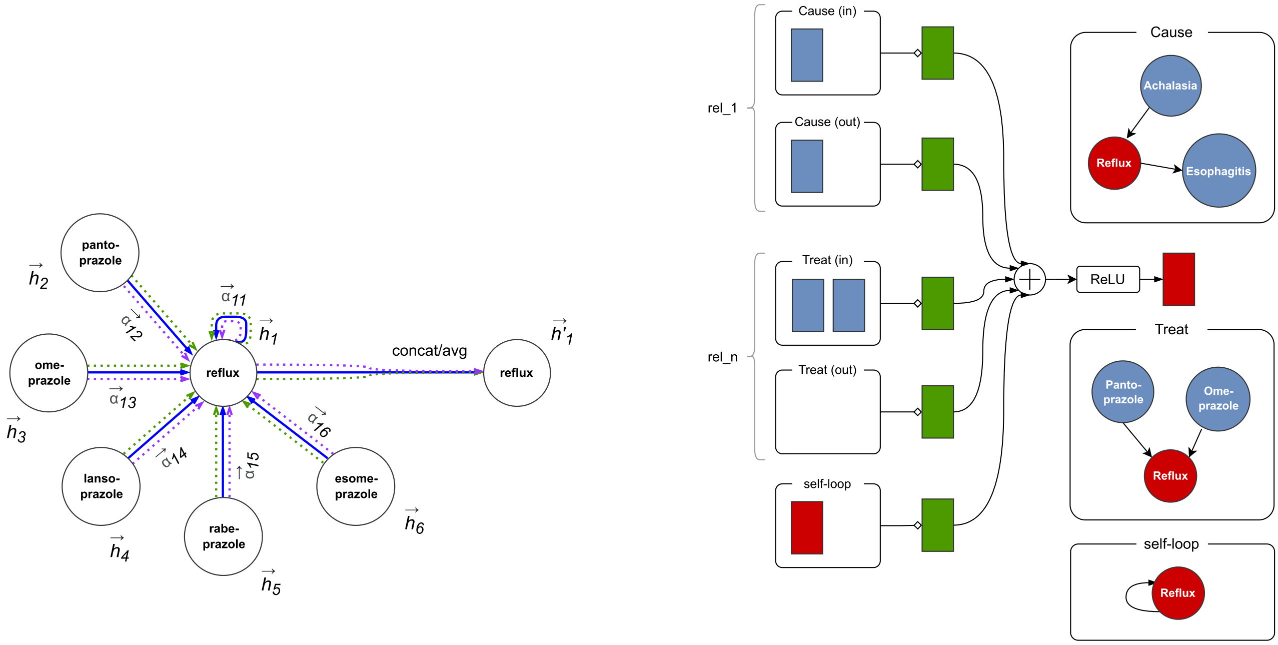


Fig. 4 Our event-augmented summarization framework. The summary is generated by attending both the event graph and the input document Figure taken from [19]

Graph Neural Network

Subsequently, the graph G is passed to a GNN. To assess the impact of edge features and broadly compare all the key graph representation learning techniques available in the literature, we explore both non-relation-aware and relation-aware architectures (sketched in Fig. 5). Indeed, the demand for processing edge-featured graphs is quite common in biomedical tasks. For example, let’s assume that the node “pantoprazole” is connected to “reflux”: the edge type—“treat” or “cause”—can utterly change the meaning of the relation. It is clear that, in such a situation, edge features can be at least as significant as those of nodes. On the other side, traditional GNNs represent structural links through binary adjacency matrices and cannot handle multi-relational graphs



(a) Illustration of multi-head attention (with K=3 heads) by node 1 on its neighborhood. Each arrow represents an independent attention computation.

(b) Diagram showing the update of a single event-graph node (red) in the R-GCN model.

Fig. 5 **a** Illustration of multi-head attention (with $K = 3$ heads) by node 1 on its neighborhood. Each arrow represents an independent attention computation. **b** Diagram showing the update of a single event-graph node (red) in the R-GCN model. GAT and R-GCN architectures used for event graph encoding

equipped with additional edge-type information. Ergo, our work evaluates the presence or absence of benefits due to the consideration of the edge type for summarization purposes, based on current GNN contributions.

Graph Attention Network

We adopt a Graph Attention Network (GAT) variant introduced in [39], working with a self-attention setup where N independent heads are calculated and concatenated before a residual connection is applied. Fundamentally, each node embedding $\hat{v}_i$ is obtained from a weighted average of its neighboring nodes $\mathcal{N}(v_i)$:

$$\alpha^n_{i,j} = \frac{\exp\left((\mathbf{W}_{1,n}\mathbf{v}_i)^\top \mathbf{W}_{2,n}\mathbf{v}_j\right)}{\sum_{z\in\mathcal{N}(v_i)} \exp\left((\mathbf{W}_{1,n}\mathbf{v}_i)^\top \mathbf{W}_{2,n}\mathbf{v}_z\right)}, \tag{1}$$

$$\hat{\mathbf{v}}_i = \mathbf{v}_i + \|_{n=1}^{N} \sum_{j\in\mathcal{N}(v_i)} \alpha^n_{i,j}\mathbf{W}_{0,n}\mathbf{v}_j, \tag{2}$$

where $\alpha^n_{i,j}$ is the attention mechanism tied to the n-th attention head, applied to node v_i and node v_j. W_* are trainable parameters.

Edge-Aware Graph Attention Network

Edge-Aware Graph Attention Networks (EGATs) [57] are a variant of GATs with an edge-type-aware message passing. The attention weights $\alpha^n_{i,j}$ are not only influenced by the features of the two nodes $\mathbf{v}_i$ and $\mathbf{v}_j$, but also by the features of the edge connecting them $\mathbf{e}_{i,j}$. We represent the latter through edge-type one-hot embeddings. The node representation learning process is the following:

$$t^n_{i,j} = \text{LeakyReLU}\left(\mathbf{a^n}^T\left[\mathbf{W^n}\mathbf{v}_i \parallel \mathbf{W^n}\mathbf{v}_j \parallel \mathbf{W^n_r}\mathbf{e}_{i,j}\right]\right), \tag{3}$$

$$\alpha^n_{i,j} = \frac{\exp\left(t^n_{i,j}\right)}{\sum_{z\in\mathcal{N}(v_i)} \exp\left(t^n_{i,z}\right)}, \tag{4}$$

$$\hat{\mathbf{v}}_i = \|_{n=1}^{N} \sum_{j\in\mathcal{N}(v_i)} \text{LeakyReLU}\left(\sum_{j\in\mathcal{N}(v_i)} \alpha^n_{i,j}\mathbf{W}^n\mathbf{v}_j\right), \tag{5}$$

where $[\ldots \parallel \ldots]$ indicates a concatenation and $\mathbf{a^n}$ a learnable weight vector.

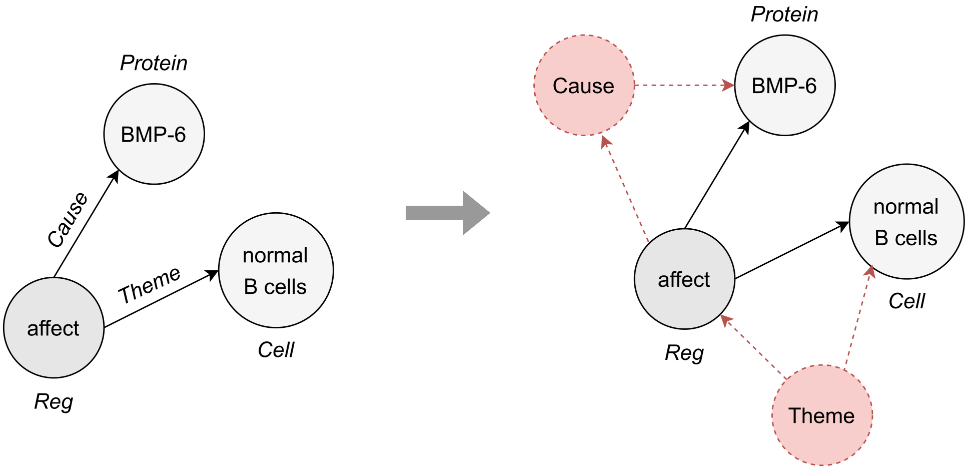


Fig. 6 Example of Levi transformation on an event-graph. Red nodes and edges are added as a kind of graph rewiring

Relational Graph Convolutional Network

Relational Graph Convolutional Networks (R-GCNs) [58] are an extension of graph convolutional networks (GCNs) capable of modeling multi-relational data. Intuitively, relation-specific transformations are introduced, depending on the type and direction of an edge:

$$\hat{\mathbf{v}}_i = \mathrm{ReLU}\left(\sum_{r\in R}\sum_{j\in\mathcal{N}(v_i)^r}\frac{1}{|\mathcal{N}(v_i)^r|}\mathbf{W}_r\mathbf{v}_j + \mathbf{W}_0\mathbf{v}_i\right), \tag{6}$$

where $\mathcal{N}(v_i)^r$ denotes the set of neighbor indices of node i under relation $r \in R$. A central issue with applying Eq. (6) is the exponential growth in parameters as the number of relation types increases. This easily brings to large-size models and overfitting on rare relations. Centrally, R-GCNs treat edge types as class labels, which indicates that edges cannot include continuous attributes.

Levi Graph Transformation and Graph Attention Network

To ponder edge features, instead of modifying the model architecture by replacing the GAT module with an R-GCN or EGAT, we can transform the input graph into its equivalent Levi graph [59]. Similarly to [32, 60, 61], each edge $e_{i,j}$ is turned into an additional node directly connected to its original linking nodes v_i and v_j (Fig. 6). The new edge set contains an edge for every $<$ node, edge $>$ pair in the original graph. We end up with an unlabeled directed graph (bipartite by definition) without the risk of parameter explosion. Edges are represented and initialized in the same way as nodes, with features given by their type description. Using this strategy, the GNN naturally generates hidden states even for edges.

Decoder

The decoder uses a multi-layer unidirectional LSTM that generates summary tokens recurrently, exploiting at each time step t the graph and the document context vectors c^v_t (Eq. 7) and c_t (Eq. 9).

Attending to the Graph

The graph context vector is computed based on the decoder hidden state s_t:

$$\mathbf{c}^v_t = \sum_i a^v_{i,t}\hat{\mathbf{v}}_i, \tag{7}$$

where $a^v_{i,t}$ denotes the attention mechanism from [62] corresponding to the ith node at time step t:

$$a^v_{i,t} = \mathrm{softmax}\left(\mathbf{u}_0^T \tanh\left(\mathbf{W}_3\mathbf{s}_t + \mathbf{W}_4\hat{\mathbf{v}}_i\right)\right). \tag{8}$$

u_* are also trainable parameters.

Attending to the Document

Similarly, the document context vector is calculated over input tokens by considering c^v_t and encoder hidden states h_k:

$$\mathbf{c}_t = \sum_k a_{k,t}\mathbf{h}_k, \tag{9}$$

where $a_{k,t}$ denotes the attention corresponding to the k-th input document token at time step t:

$$a_{k,t} = \mathrm{softmax}\left(\mathbf{u}_1^T \tanh\left(\mathbf{W}_5\mathbf{s}_t + \mathbf{W}_6\mathbf{h}_k + \mathbf{W}_7\mathbf{c}^v_t\right)\right). \tag{10}$$

Token Prediction

The decoder hidden state s_t is concatenated to the graph and document context vectors, expressing the salient content coming from both sources. This final representation is used to determine the probability distribution of the vocabulary vocab at time step t:

$$P_{\text{vocab},t} = \text{softmax}(\mathbf{W}_{\text{out}}\,[\mathbf{s}_t \| \mathbf{c}_t \| \mathbf{c}^v_t]). \tag{11}$$

We also include a copy mechanism as in [32] to check out the embedding of the token generated at the previous time step y_{t-1}:

$$P_{\text{copy},t} = \sigma(\mathbf{W}_{\text{copy}}[\mathbf{s}_t \| \mathbf{c}_t \| \mathbf{c}^v_t \| \mathbf{y}_{t-1}]). \tag{12}$$

$P_{\text{copy},t} \in [0, 1]$ is used as a soft switch to decide between generating a token from the vocabulary by sampling from $P_{\text{vocab},t}$, or copying a token from the input sequence by sampling from the attention distribution $a_{k,t}$. The probability of generating the token w at time t is given by:

$$P_t(w) = P_{\text{copy},t} P_{\text{vocab},t}(w) + \left(1 - P_{\text{copy},t}\right) \sum_{k:w_k=w} a_{k,t}. \tag{13}$$

Training Objective

We employ a negative log-likelihood loss function between the generated summary $\hat{\mathbf{y}}$ and the ground-truth $\mathbf{y}$:

$$\mathcal{L} = -\frac{1}{|D|} \sum_{(\mathbf{y},\mathbf{x}) \in D} \log p_\theta(\mathbf{y} \mid \mathbf{x}, G), \tag{14}$$

where $\mathbf{x}$ are the source documents and $\mathbf{y}$ and are the target summaries from training set D, G is the graph constructed from x, and $\theta = \{\mathbf{W}_*, \mathbf{u}_*\}$ is the set of the model trainable parameters.

Pseudocode

Algorithm 1 provides a concise explanation of the autoregressive generation of summary tokens y, starting from input document x.

Algorithm 1 Summary generation

```
Input: input document tokens $x = \{x_1, x_2, ...., x_n\}$
Output: generated summary tokens $y = \{y_1, y_2, ...., y_m\}$
$h^* = \{h^*_1, h^*_2, ...., h^*_n\} \leftarrow scibert.encode(x)$
$h = \{h_1, h_2, ...., h_n\} \leftarrow lstm.encode(h^*)$
$.a2 \leftarrow dem.parser(x)$
$G = (V, E) \leftarrow construct\_G(.a2)$
$\mathbf{v} = \{\mathbf{v_1}, \mathbf{v_2}, ...., \mathbf{v_k}\} \leftarrow [h \| s_a]$
if $GNN = EGAT$ or $GNN = RGCN$ then
    $\hat{\mathbf{v}} = \{\hat{\mathbf{v}}_1, \hat{\mathbf{v}}_2, ...., \hat{\mathbf{v}}_k\} \leftarrow GNN(\mathbf{v}, E)$
else
    $\hat{\mathbf{v}} = \{\hat{\mathbf{v}}_1, \hat{\mathbf{v}}_2, ...., \hat{\mathbf{v}}_k\} \leftarrow GNN(\mathbf{v})$
end if
$t = 0;\ y_1 = SOS$
while $y_t \neq CLS$ do
    $\mathbf{c}^v_t = graph\_attention.layer(\hat{\mathbf{v}})$
    $\mathbf{c}_t = doc\_attention.layer(h)$
    $y_t \leftarrow lstm.decode(\mathbf{c}^v_t, \mathbf{c}_t)$
    $t \mathrel{+}= 1$
end while
```

Experimental Setup

Dataset

We evaluate EASumm on the CDSR dataset [18], a publicly available corpus designed for assessing the automated generation of lay language summaries from biomedical scientific reviews. Besides creating accurate and factual summaries, this benchmark also requires a joint style transition from the original language of healthcare professionals to that of the general public. By imposing high abstraction and biomedical explanation constraints, CDSR is an ideal testbed. The training, validation, and test sets contain 5178, 500, and 999 samples. The documents can be downloaded directly from the Cochrane Database of Systematic Review[1] As for EE, each source document was split into a set of sentences and passed to DEM; the results were saved in standoff $.a*$ files. Statistics about the total numbers of events, entities, and triggers extracted by DEM are detailed in Table 9.

Training Details and Parameters

All experiments were run using a single NVIDIA GeForce RTX 3090. We used the cased version of SciBERT to extract

[1] https://www.cochranelibrary.com/cdsr/reviews (accessed Nov. 4, 2022). We point out to the reader that the authors claim a slightly higher number of examples (5195, 500, and 1000), then removed from the official repository.

Table 1 Final picked values for model hyperparameters

Hyperparameters	
LSTM input word embedding size	128
LSTM hidden embedding size	256
LSTM number of layers	2
Dropout rate	0.1
Learning rate	1×10^{-3}
Optimizer	AdamW (0.9 β_1, 0.999 β_2, 0.5 w. decay)
Decoding strategy	Beam Search
Number of beams	5
GAT	
Number of self-attention heads	4
Hidden size	556
Node size	556
EGAT	
Hidden size	556
Node size	556
Edge size	10
Number of layers	2
R-GCN	
Hidden size	556
Node size	556
Number of relations	10
Regularization	Block-diagonal-decomposition (4 blocks)
Aggregation scheme	Mean

token embeddings. Hyperparameters are listed in Table 1. We implemented RGCN and EGAT with Pytorch Geometric [63], while GAT is drawn on [43]. We used the version of DEM pretrained on the MLEE task[2] [64]—the EE benchmark linked to the biomedical domain most aligned to CDSR based on empirical tests (see "Event extraction dataset selection").

Material

For replication purposes, the code and the dataset are publicly available at https://github.com/disi-unibo-nlp/easumm.

Baseline Methods and Comparisons

We conduct comprehensive ablation studies by testing different EASumm variants (hereinafter shortened as EAS), which we denote through the suffix, with "−" symbolizing a module exclusion and "+" an addition/substitution:

- −G stands for the graph encoder exclusion;
- + RB indicates the adoption of RoBERTa [56] instead of SciBERT to generate source document tokens embeddings;
- −Type refers to the node type exclusion during the initialization;
- + EGAT + RGCN specify the adoption of EGAT and RGAT, respectively, in replacement of GAT;
- + BIP suggests the employment of the Levi transformation on GAT-processed event graphs to treat nodes and edges equally.

For a comparative analysis, we experiment with two extractive methods:

- *Oracle extractive*: it creates an oracle summary by selecting the set of sentences in the document that generates the highest ROUGE-2 score with the ground-truth summary (i.e., syntactic match upper bound);
- *BERT* [55]: inter-sentence encoder with classification head, supervised through an Oracle extractive signal;

and two abstractive methods:

- *Pointer generator* [65]: standard seq2seq model with a pointer network that allows both copying words from the source and generating new words from a fixed vocabulary;
- *BART* [25]: full-transformer pretrained on large corpora by reconstructing text after a corruption phase with an arbitrary noising function. We also take into account a variant with additional pretraining steps on PubMed to compensate for the limited training data. Specifically, we use the PMC articles dataset,[3] containing 300K PubMed abstracts.

Evaluation

Quantitative Analysis

As done in [18], we use ROUGE [66] to evaluate the summarization performance. ROUGE-n quantifies the overlap of n-grams between the model-generated summary and the human-generated reference summary, and ROUGE-L measures the longest matching sequence of words using the longest common subsequence. We report the

[2] http://nactem.ac.uk/MLEE.

[3] https://www.kaggle.com/cvltmao/pmc-articles.

ROUGE-1, ROUGE-2 and ROUGE-L scores computed using `pyrouge`.[4]

Given the supplementary scientific → public language translation objective of the CDSR task, other than informativeness, we are interested in measuring the ease with which a reader can understand a passage, defined as readability. We use three standard metrics for this goal: Flesch-Kincaid grade level [67], Gunning fog index [68], and Coleman-Liau index [69]. Their equations are as follows:

- **Flesch-Kincaid grade level**

$$0.39\left(\frac{\text{total words}}{\text{total sentences}}\right) + 11.8\left(\frac{\text{total syllables}}{\text{total words}}\right) - 15.59, \tag{15}$$

- **Gunning fog index**

$$0.4\left[\left(\frac{\text{words}}{\text{sentences}}\right) + 100\left(\frac{\text{complex words}}{\text{words}}\right)\right], \tag{16}$$

  where complex words are those words with three or more syllables.

- **Coleman–Liau index**

$$0.0588L - 0.296S - 15.8, \tag{17}$$

  where L and S are the average numbers of letters and sentences per 100 words, respectively.

All these evaluation metrics are computed using `textstat`[5] and estimate the years of formal education a person needs to understand the text. Lower scores indicate that the text is easier to read; for instance, scores of 13–16 correspond to college-level reading ability in the United States education system.

Qualitative Analysis

Automatic evaluation metrics for summarization are not able to grasp all the desired quality dimensions, particularly in highly abstractive settings grounded on semantics [70]. To fill this gap, we run an in-depth human evaluation study to analyze proper text properties and identify primary error sources. We randomly sample 50 CDSR test set instances and engage three native or fluent English speakers with biomedical expertise (average age: 24.6 years old; average time for completion: 2 h; education level: 1 PhD and 2 master students; no compensation). Selection criteria guarantee that our annotators are representative of the college-educated lay public. Precisely, we presented each human rater with the source document, the inferred summary, and the reference summary. Then, we asked raters to judge the prediction along three quality criteria with a Likert scale from 1 (worst) to 5 (best).

- *Informativeness*. Does the summary supply enough necessary content coverage from the input article?
- *Fluency*. Does the text progress naturally? Is it grammatically correct (e.g., no fragments and missing components) and coherent whole?
- *Understandability*, CDSR-related [18]. Is the summary more effortless to understand than the source?

We even invite evaluators to binary label whether summaries contain any of the following classes of unfaithful errors: (1) *Hallucination*, fabricated content not present in the input; (2) *Deletion or substitution*, erroneously missing or edited elements (e.g., entities with altered semantic role); (3) *Repetitiveness*, repeated fragments. Complete guidelines are in “Human evaluation guideline”.

Results

Automated Summary Evaluation

Evaluation on full dataset

Table 2 exhibits the results of our presented models compared to baseline methods. Notably, EASUMM gives better ROUGE scores than all its variants. The positive effect of the event graph is motivated by the performance drop associated with EASUMM−G. Features obtained via a domain-coherent language model like SciBERT contribute to superior results than RoBERTa. Further, the graph encoder in the RoBERTa implementation does not furnish any progress over the solution without it. Type-augmented node initialization techniques show clear advantages, confirming our hypotheses on the usefulness of domain-specific and semantic text augmentations pulled by DEM. EASUMM significantly outperforms BERT, pointer generator, and plain Bi-LSTM architectures but does not beat BART (quality gap of ≈6 ROUGE points), despite greater readability on average. This behavior suggests a future direction of developing event-driven models on top of a large pretrained encoder-decoder model such as BioBART [71]. By contra, we emphasize the lightness of our final model compared to $\text{BART}_{BASE/LARGE}$, which counts 8 M trainable parameters instead of 139 M/406 M (up to 50x fewer weights). Finally, we mention the importance of expanding training data with more biomedical corpora.

[4] https://github.com/bheinzerling/pyrouge.

[5] https://pypi.org/project/textstat/, version 0.72.

Table 2 Automated evaluation on the full test set of CDSR with ROUGE and readability metrics

Model	ROUGE-1	ROUGE-2	ROUGE-L	Flesch-Kincaid	Gunning	Coleman-Liau
Oracle extractive	**53.56**	**25.54**	**49.56**	14.85	13.45	16.13
BERT	26.60	11.11	24.59	**13.44**	**13.26**	**14.40**
Pointer generator	38.33	14.11	35.81	16.36	15.86	15.90
$\text{BART}_{\text{BASE}}$	51.39	20.81	48.56	14.31	18.13	**14.00**
$\text{BART}_{\text{LARGE}}$	52.53	**21.83**	49.75	13.59	14.16	14.45
$\text{BART}_{\text{LARGE}}$+PubMed	**52.66**	21.73	**49.97**	**13.30**	**13.80**	14.28
Ours						
EAS-G+RB	44.23	18.03	41.68	14.05	17.86	14.05
EAS+RB	44.12	17.82	41.60	13.57	17.29	13.77
EAS-G	44.68	17.95	42.25	12.41	16.76	12.82
EAS-Type	45.41	18.36	42.99	**12.14**	16.40	12.91
EAS	**46.30**	**18.73**	**43.78**	12.42	16.68	13.06

Top: extractive models. Middle: abstractive models. Bottom: our event-augmented abstractive models. The best scores for each model type are boldened

Table 3 ROUGE performance on four testset subsets, depending on the minimum number of extracted events per sentence (EEpS)

EEpS	Model	R-1	R-2	R-L
> 0.4	$\text{BART}_{\text{BASE}}$	49.55	18.89	46.60
	EAS-G+RB	44.45	17.65	41.13
	EAS+RB	45.97	17.88	42.95
	EAS-G	45.41	17.60	42.38
	EAS	47.29 ↑	18.50 ↑	44.60 ↑
> 0.3	$\text{BART}_{\text{BASE}}$	49.75	19.12	46.70
	EAS-G+RB	44.53	17.09	41.54
	EAS+RB	44.97	16.96	42.09
	EAS-G	43.87	16.77	41.14
	EAS	46.77 ↑	17.95 ↓	44.14 ↑
> 0.2	$\text{BART}_{\text{BASE}}$	49.81	19.31	46.84
	EAS-G+RB	44.15	16.92	41.34
	EAS+RB	44.16	16.86	41.44
	EAS-G	43.78	16.79	41.07
	EAS	46.10 ↓	18.19 ↓	43.42 ↓
> 0.1	$\text{BART}_{\text{BASE}}$	50.77	20.23	47.81
	EAS-G+RB	44.45	17.55	41.73
	EAS+RB	44.48	17.41	41.82
	EAS-G	44.67	17.50	42.06
	EAS	46.18	18.39	43.51

↑ and ↓ symbols denote the score increase and decrease w.r.t. the previous subset, respectively

Evaluation on Subsets

As documented in "Event extraction dataset selection", the number of events extracted in each document may be contained, leading to sparse graphs with few nodes. Thus, we suspect that the graph encoder contribution could be capped, expecting a more noticeable performance gap regarding EASumm−G for those documents containing a larger number of events extracted per sentence (abbreviated as EEpS). Following this line of thought, we assembled four subsets where source documents have an EEpS greater than 0.1, 0.2, 0.3, and 0.4. Table 3 reveals the ROUGE scores on each of the four subsets for the different model variants and $\text{BART}_{\text{BASE}}$. As EEpS increases, the performance gap between the solutions with and without graph encoder widens, proving our speculation. We can also notice how the EASumm performance gets closer to $\text{BART}_{\text{BASE}}$. Given the linear relationship between EEpS and the ROUGE gap—emerged from our empirical experiments, we conjecture that the deficit can be wiped out entirely with event extraction models pretrained on sentences conveying topics more aligned to the CDSR ones (Table 4).

The Impact of Relations During Graph Representation Learning

As pointed out in Tables 5 and 6 from the point of view of ROUGE scores, the solutions adopting EGAT and Levi transformation yield the best results, underlining the major limit of RGCNs, which is not being able to process custom representations for edge features. Distinctly, EGAT is the graph modeling technique that most benefits from higher EEpS; by contra, the other two methods are characterized by fluctuating ROUGE scores with respect to EEpS. In terms of readability, EGAT dominates all three metrics, followed by RGCN and Levi transformation. Surprisingly, we couldn't improve the original solution that ignores relation types when encoding the graph. However, by looking at the distribution of relation types (see Table 4), we notice that it is extremely uneven. In particular, almost all nodes are linked by either an *Instrument* or *Theme* edge. The other 8 relation

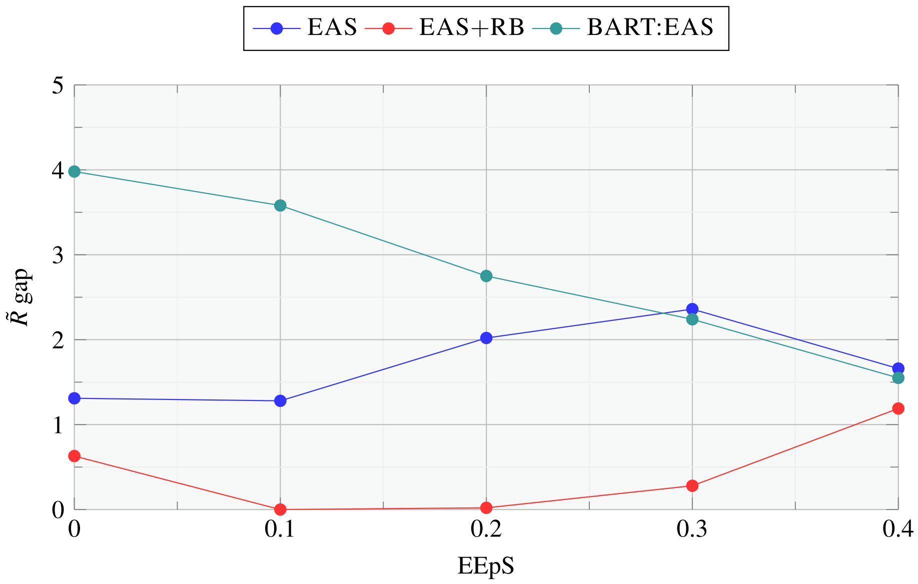


Fig. 7 Performance gap—measured as $\tilde{R}$ (average of ROUGE-1, ROUGE-2, and ROUGE-L)—between event-augmented models and the number of extracted events per sentence (EEpS). With EAS and EAS+RB, the gap is measured w.r.t. variants without the graph encoder. BART:EAS tracks the gap between the fine-tuned BART$_{\text{BASE}}$ and EAS

Table 4 Absolute frequencies of event extraction relation types identified in the whole dataset

Instrument	Theme	Cause	ToLoc	Participant	FromLoc	AtLoc
11,387	10,812	152	20	8	2	5

Table 5 Automated evaluation on the full test set of CDSR with ROUGE and readability metrics for EASumm models with relation-aware graph representation learning

Model	ROUGE-1	ROUGE-2	ROUGE-L	Flesch-Kincaid	Gunning	Coleman-Liau
EAS+ EGAT	45.14	18.16	42.73	**12.21**	**16.32**	**12.71**
EAS+ RGCN	44.79	17.74	42.36	12.40	16.59	13.02
EAS+ BIP	45.35	17.96	42.90	12.26	16.51	12.98
EAS	**46.30**	**18.73**	**43.78**	12.42	16.68	13.06

The best scores for each model type are boldened

Table 6 Link between ROUGE performance and minimum number of extracted events per sentence (EEpS) in the case of relation-aware graph representation learning

EEpS	Model	R-1	R-2	R-L
> 0.4	EAS+EGAT	46.35	18.35	43.62
	EAS+RGCN	45.28	18.11	42.25
	EAS+BIP	45.94	17.16	42.91
> 0.3	EAS+EGAT	45.33	17.33	42.77
	EAS+RGCN	44.49	17.04	41.78
	EAS+BIP	44.71	16.78	41.76
> 0.2	EAS+EGAT	45.71	17.90	43.14
	EAS+RGCN	44.73	17.10	42.05
	EAS+BIP	45.17	17.14	42.42
> 0.1	EAS+EGAT	45.29	18.03	42.76
	EAS+RGCN	45.20	17.48	42.57
	EAS+BIP	45.54	17.62	42.89

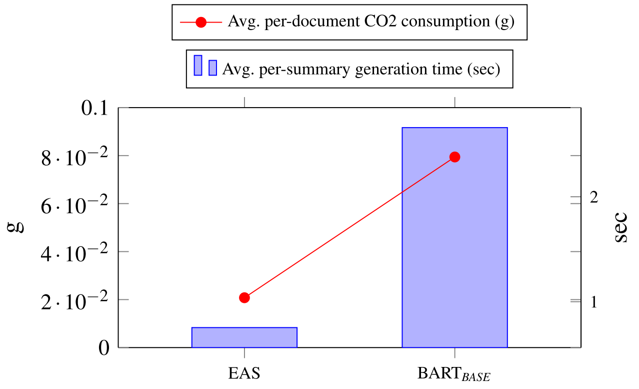


Fig. 8 Comparison between EAS with BiLSTMs (ours) and BART$_{BASE}$ in terms of inference time and carbon footprint on the CDSR test set

types don't seem to have a relevant impact; therefore, we can easily understand why this negatively affects the potential contribution of edge-aware solutions (Fig. 7).

Table 7 Average human evaluation scores on informativeness (Inf.), fluency (Flu.), and understandability (Und.) (1-to-5), with error percentages for hallucination (Hal.), deletion or substitution (Del./Sub.), and repetitiveness (Rep.)

Inf.	Flu.	Und.	Hal.	Del./Sub.	Rep.
3.16	3.4	3.44	18%	35%	34%

Inference Time and CO_2 Impact

As motivated by the latest graph-enhanced summarizers [41, 44, 45], we do not utilize an encoder-decoder architecture based on pretrained language models due to their environmental cost and computational requirements. Although our model only uses BiLSTM and GNN structures (graph-LSTM), experimental results prove that it still achieves competitive performance. Furthermore, compared to SOTA generators like BART, BiLSTM is a lightweight architecture in terms of size, inference time, and CO_2 impact—tracked with CodeCarbon [72] (Fig. 8). Indeed generating a single summary with $\text{BART}_{\text{BASE}}$ requires 2.65 seconds and produces 7.9×10^{-2} grams of CO_2, while EASumm needs just 0.75 s and consumes 7.9×2^{-2} grams of CO_2. The adoption of Green NLP technology can revolutionize the way we use AI to understand and address environmental issues [73].

Human Evaluation

Table 7 portrays the results of human evaluations. The average inter-rater agreement is 0.61 (Kendall's coefficient $\in [-1, 1]$ indicating low to high association), a good score considering the subjectivity of the rating task. For full transparency, we publicly release the results of our human evaluation.[6] Besides the need for larger-scale studies, this work delivers helpful preliminary evidence. EASumm obtains suitable scores in fluency and understandability. Deletion and substitution in verbalized facts appear to be the most frequent error type, together with repetitiveness. After inspection, we find several utterances with swapped entities not belonging to event mentions, thus not attributable to a non-effectiveness of event injection. Low hallucinations testify to the advantage of leveraging event graph representations. With a closer look, we observe that human-written summaries also include a non-trivial amount of commonsense and world knowledge not mentioned by the input article. For example, for a source document discussing "spironolactone", the human writer may add "used since the 1960s" in the summary. Consequently, we invite the reader to reflect on attributing low factuality scores to generative models, weighting them against the dataset's properties.

Conclusion

We introduced EASumm, the first abstractive summarization model augmenting source documents with explicit, structured medical evidence extracted from them, thereby concretizing a tandem text-graph architecture. We demonstrated the significant positive influence of biomedical event extraction for summarization, allowing a model to better distinguish semantics and lexical surface. Indeed, we showed improvements in ROUGE and readability scores, observing a strong connection between the summer quality and (1) the number of events extracted from the input document, (2) the enhanced node features initialization considering both domain-specific pretrained language models and entity types. Contrary to expectations, event-graph representation learning does not benefit from the awareness of the relation type. The motivation is to be found in the task-driven nature of event extraction and in the poor capacity of current graph neural networks in managing multi-relational graphs. Although the numerous newly introduced graph-LSTM models combined with structured knowledge, we establish that these architectures are far from being competitive with generative transformer-based solutions like BART.

Future Directions

Based on our findings, we suggest nine promising future research directions:

1. using large pretrained encoder-decoder transformers to replace graph-LSTMs architectures [74];
2. increasing the number of the events by merging available biomedical benchmarks thanks to generative event extraction approaches and non-discriminative architectures [75];
3. increasing the size of the document-level graphs by combining events with linguistically-grounded abstract meaning representations [74];
4. exploiting multimodal text-graph alignments with metric learning techniques as in [76–78];
5. exploring end-to-end event extraction and document summarization;
6. discovering new connections between nodes useful for increasing summarization performance (i.e., dynamic event graph construction), conveying techniques like random perturbation [79, 80] and iterative deep graph learning [81];
7. performing node relevance scoring supported by term weighting [82] and/or perplexity metrics [12];

[6] https://github.com/disi-unibo-nlp/easumm/blob/master/sn_human_evaluations.xlsx.

Table 8 CDSR average number of words (N. words), sentences (N. sents), and readability

Document	Set	N. words	N. sents	Readability
Source	Train	644	26	16.43
	Val	643	26	16.60
	Test	653	27	16.45
Target	Train	349	16	15.15
	Val	348	16	15.20
	Test	353	16	15.22

8. devising transfer-learning methods [75, 83, 84] across multiple biomedical fields;
9. utilizing continuous (semantic) edge features within the graph neural network;
10. introducing additional loss functions based on reinforcement learning and semantic-driven rewards [74];
11. pushing the interaction and mutual influence between graph and text encoders;
12. injecting subgraphs fetched from external structured memories using dense representations [85], devising retrieval-enhanced language models [86].

Appendix A: Section Title of First Appendix

Dataset Statistics

We report additional statistics for each source and target document in CDSR (Table 8). Note: a readability score is calculated by averaging the results of the metrics described in "Evaluation".

Table 9 CDSR event extraction results using distinct versions of DeepEventMine pretrained on MLEE [64], CG13 [87] and GE13 [88] tasks

Task	Set	N. evs.	N. trigs.	N. args.
MLEE	Train	2.63	2.31	2.78
	Val	2.54	2.20	2.73
	Test	2.70	2.42	2.84
CG13	Train	2.13	1.95	2.30
	Val	2.02	1.85	2.19
	Test	2.12	1.95	2.30
GE13	Train	0.05	0.05	0.05
	Val	0.06	0.06	0.07
	Test	0.07	0.06	0.06

We report the average number of events (N. evs.), triggers (N.trigs.) and arguments (N. args.) extracted from training, validation and test samples in each source document

Event Extraction Dataset Selection

Table 9 provides statistics on the effectiveness of the three DEM models trained on the available biomedical datasets with the largest number of annotations and ontological targets [10]. MLEE stands out as the EE task most related to CDSR topics.

Table 10 Quality aspect scales and value-level explanations

	Informativeness:
1	Not relevant to the article
2	Partially relevant and misses the main point of the article
3	Relevant, but misses the main point of the article
4	Successfully captures the main point of the article but some relevant content is missing
5	Successfully captures the main point of the article
	Fluency:
1	Summary is full of garbage fragments and is hard to understand
2	Summary contains fragments, missing components but has some fluent segments
3	Summary contains some grammar errors but is in general fluent
4	Summary has relatively minor grammatical errors
5	Fluent summary
	Understandability:
1	Source is easier to understand than the summary
2	Summary is as understandable as the source
3	Summary is easier to understand than the source but it is partially written in the language of healthcare professionals
4	Summary is easier to understand than the source but contains some terms from the language of healthcare professionals
5	Summary is easier to understand than the source and is written in the language of the general public

Human Evaluation Guideline

Table 10 explains each Likert scale score meaning for the assessed quality criteria, so as to minimize annotation ambiguity and subjectivity. We believe this is important to obtain comparable results and work towards an objective and replicable human evaluation.

Funding Open access funding provided by Alma Mater Studiorum - Università di Bologna within the CRUI-CARE Agreement.

Declarations

Conflict of interest The authors declare no conflict of interest.

References

1. Pinker S. The language instinct. New York: William Morrow & co; 1994.
2. Brown T, Mann B, Ryder N, Subbiah M, et al. Language models are few-shot learners. In: Larochelle H, Ranzato M, Hadsell R, Balcan MF, et al., editors. Advances in Neural Information Processing Systems, vol. 33. Virtual: Curran Associates Inc; 2020. p. 1877–901. https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf
3. Bender EM, Gebru T, McMillan-Major A, Shmitchell S. On the dangers of stochastic parrots: can language models be too big? In: Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency (FAccT '21). New York, NY, USA: Association for Computing Machinery; 2021. p. 610–23.
4. Zhou C, Neubig G, Gu J, Diab M, Guzmán F, Zettlemoyer L, Ghazvininejad M. Detecting hallucinated content in conditional neural sequence generation. In: ACL/IJCNLP (Findings). Findings of ACL, vol. ACL/IJCNLP 2021. Bangkok: Association for Computational Linguistics; 2021. pp. 1393–404.
5. Zhang WE, Sheng QZ, Alhazmi A, Li C. Adversarial attacks on deep-learning models in natural language processing: a survey. ACM Trans Intell Syst Technol. 2020;11(3):24–12441. https://doi.org/10.1145/3374217.
6. Moradi M, Ghadiri N. Text summarization in the biomedical domain 2019. arXiv preprint arXiv:1908.02285
7. Frisoni G, Moro G, Carbonaro A. Learning interpretable and statistically significant knowledge from unlabeled corpora of social text messages: A novel methodology of descriptive text mining. In: DATA 2020 - Proceedings of 9th International Conference Data Science, Technology and Application. SciTePress, Virtual; 2020. p. 121–34. https://www.scopus.com/inward/record.uri?eid=2-s2.0-85092009636 &partnerID=40 &md5=27541a3b46d782bb7984eed8ba7fa8a3
8. Frisoni G, Moro G. Phenomena explanation from text: unsupervised learning of interpretable and statistically significant knowledge. In: DATA (Revised Selected Papers), vol. 1446. Cham: Springer; 2020. p. 293–318. https://doi.org/10.1007/978-3-030-83014-4_14. https://www.scopus.com/inward/record.uri?eid=2-s2.0-85113292013 &doi=10.1007%2f978-3-030-83014-4_14 &partnerID=40 &5=33fa92fd1f11dff84de31aac3729917a
9. Frisoni G, Moro G, Carbonaro A. Towards rare disease knowledge graph learning from social posts of patients. In: RiiForum. Cham: Springer; 2020. p. 577–89. https://doi.org/10.1007/978-3-030-62066-0_44. https://www.scopus.com/inward/record.uri?eid=2-s2.0-85102640128 &doi=10.1007%2f978-3-030-62066-0_44 &partnerID=40 &md5=7b08bda5b0f9de00d4e5acdaccfe7707
10. Frisoni G, Moro G, Carbonaro A. A survey on event extraction for natural language understanding: riding the biomedical literature wave. IEEE Access. 2021;9:160721–57. https://doi.org/10.1109/ACCESS.2021.3130956.
11. Colon-Hernandez P, Havasi C, Alonso JB, Huggins M et al. Combining pre-trained language models and structured knowledge. CoRR. arXiv:2101.12294 2021.
12. Yasunaga M, Ren H, Bosselut A, Liang P, Leskovec J. QA-GNN: reasoning with language models and knowledge graphs for question answering. CoRR. arXiv:2104.06378 2021.
13. Zhang Z, Wu Y, Zhao H, Li Z et al. Semantics-aware bert for language understanding. In: Proceedings of the AAAI conference on artificial intelligence, New York, USA, vol. 34; 2020. p. 9628–35.
14. Domeniconi G, Semertzidis K, López V, Daly EM, et al. A novel method for unsupervised and supervised conversational message thread detection. In: DATA 2016—Proceedings of 5th International Conference Data Science, Technology and Application. Lisbon: SciTePress; 2016. p. 43–54. https://doi.org/10.5220/0006001100430054
15. Domeniconi G, Moro G, Pagliarani A, Pasini K, et al. Job recommendation from semantic similarity of linkedin users' skills. In: ICPRAM 2016. Rome: SciTePress; 2016. p. 270–77. https://doi.org/10.5220/0005702302700277. https://www.scopus.com/inward/record.uri?eid=2-s2.0-84970039381 &doi=10.5220%2f0005702302700277 &partnerID=40 &md5=eca4633aae1e9418df034aaa5f3a6020
16. Frisoni G, Moro G, Carbonaro A. Unsupervised descriptive text mining for knowledge graph learning. In: IC3K 2020—Proceedings of 12th International Joint Conference Knowledge Discovery, Knowledge Engineering and Knowledge Management, vol. 1. SciTePress, Virtual; 2020. p. 316–24. https://www.scopus.com/inward/record.uri?eid=2-s2.0-85107113340 &partnerID=40 &md5=7a4cc3ae8a6894d1a3fff499bb4bf717
17. Bui Q-C, Sloot MAP. A robust approach to extract biomedical events from literature. Bioinformatics. 2012;28(20):2654–61. https://doi.org/10.1093/bioinformatics/bts487.
18. Guo Y, Qiu W, Wang Y, Cohen, T. Automated lay language summarization of biomedical scientific reviews. In: AAAI. AAAI Press, Virtual; 2021. p. 160–68
19. Frisoni G, Italiani P, Boschi F, Moro G. Enhancing biomedical scientific reviews summarization with graph-based factual evidence extracted from papers. In: Cuzzocrea A, Gusikhin O, van der Aalst WMP, Hammoudi S, editors. Proceedings of the 11th international conference on data science, technology and applications, DATA. Lisbon: SCITEPRESS; 2022. pp. 168–79. https://doi.org/10.5220/0011354900003269
20. Liu Y, Lapata M. Text summarization with pretrained encoders. In: Inui K, Jiang J, Ng V, Wan X, editors. EMNLP/IJCNLP. Hong Kong, China: Association for Computational (1); 2019. p. 3730–40.

21. Dong L, Yang N, Wang W, Wei F, et al. Unified language model pre-training for natural language understanding and generation. In: Wallach H, Larochelle H, Beygelzimer A, d' Alché-Buc F, et al. editors. Advances in neural information processing systems, vol. 32. Vancouver: Curran Associates, Inc.; 2019. https://proceedings.neurips.cc/paper/2019/file/c20bb2d9a50d5ac1f713f8b34d9aac5a-Paper.pdf
22. Rothe S, Narayan S, Severyn A. Leveraging pre-trained checkpoints for sequence generation tasks. Trans Assoc Comput Linguist. 2020;8:264–80. https://doi.org/10.1162/tacl_a_00313.
23. Zhang J, Zhao Y, Saleh M, Liu PJ. PEGASUS: pre-training with extracted gap-sentences for abstractive summarization. CoRR. arXiv:1912.08777 2019.
24. Qi W, Yan Y, Gong Y, Liu D, et al. Prophetnet: predicting future n-gram for sequence-to-sequence pre-training. In: EMNLP (Findings). Findings of ACL, vol. EMNLP. Association for Computational Linguistics, Virtual; 2020. p. 2401–10.
25. Lewis M, Liu Y, Goyal N, Ghazvininejad M, Mohamed A, Levy O, Stoyanov V, Zettlemoyer L. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In: ACL, Association for Computational Linguistics, Virtual; 2020. p. 7871–80.
26. Moro G, Ragazzi L. Semantic self-segmentation for abstractive summarization of long legal documents in low-resource regimes. In: Thirty-Sixth AAAI conference on artificial intelligence. AAAI 2022. Virtual: AAAI Press; 2022. p. 1–9.
27. Moro G, Ragazzi L, Valgimigli L, Frisoni G, Sartori C, Marfia G. Efficient memory-enhanced transformer for long-document summarization in low-resource regimes. Sensors. 2023;23(7):1. https://doi.org/10.3390/s23073542.
28. Moro G, Ragazzi L, Valgimigli L, Freddi D. Discriminative marginalized probabilistic neural method for multi-document summarization of medical literature. In: Proceedings of the 60th annual meeting of the association for computational linguistics, vol. 1: long papers. Dublin: Association for Computational Linguistics; 2022. p. 180–89. https://doi.org/10.18653/v1/2022.acl-long.15. https://aclanthology.org/2022.acl-long.15
29. Maynez J, Narayan S, Bohnet B, McDonald RT. On faithfulness and factuality in abstractive summarization. In: ACL, Association for Computational Linguistics, Virtual 2020. p. 1906–919.
30. Pasunuru R, Bansal M. Multi-reward reinforced summarization with saliency and entailment. In: NAACL-HLT. Melbourne: Association for Computational Linguistics (2); 2018. p. 646–53.
31. Arumae K, Liu F. Guiding extractive summarization with question-answering rewards. arXiv preprint arXiv:1904.02321 2019.
32. Huang L, Wu L, Wang L. Knowledge graph-augmented abstractive summarization with semantic-driven cloze reward. In: ACL, Association for Computational Linguistics, Virtual; 2020. p. 5094–107.
33. Bender EM, Koller A. Climbing towards NLU: on meaning, form, and understanding in the age of data. In: ACL. Association for Computational Linguistics, Virtual; 2020. p. 5185–198.
34. Mihalcea R, Tarau P. Textrank: bringing order into text. 2004.
35. Wan X. An exploration of document impact on graph-based multi-document summarization. In: Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing (EMNLP '08). Honolulu, Hawaii: Association for Computational Linguistics; 2008. p. 755–62.
36. Tan J, Wan X, Xiao J. Abstractive document summarization with a graph-based attentional neural model. In: ACL (1). Vancouver: Association for Computational Linguistics; 2017. p. 1171–181.
37. Fernandes P, Allamanis M, Brockschmidt M. Structured neural summarization. In: ICLR (Poster). OpenReview.net, New Orleans, Louisiana; 2019.
38. Song L, Zhang Y, Wang Z, Gildea D. A graph-to-sequence model for amr-to-text generation. In: ACL (1). Melbourne: Association for Computational Linguistics; 2018. p. 1616–626.
39. Koncel-Kedziorski R, Bekal D, Luan Y, Lapata M, Hajishirzi H. Text generation from knowledge graphs with graph transformers. In: NAACL-HLT (1). Florence: Association for Computational Linguistics; 2019. p. 2284–293.
40. Agarwal O, Ge H, Shakeri S, Al-Rfou R. Knowledge graph based synthetic corpus generation for knowledge-enhanced language model pre-training. In: NAACL-HLT. Association for Computational Linguistics, Virtual; 2021. p. 3554–565.
41. An C, Zhong M, Chen Y, Wang D, et al. Enhancing scientific papers summarization with citation graph. In: AAAI. AAAI Press, Virtual; 2021. p. 12498–2506.
42. Fan A, Gardent C, Braud C, Bordes A. Using local knowledge graph construction to scale seq2seq models to multi-document inputs. In: EMNLP/IJCNLP (1). Hong Kong: Association for Computational Linguistics; 2019. p. 4184–194.
43. Huang L, Wu L, Wang L. Knowledge graph-augmented abstractive summarization with semantic-driven cloze reward. In: ACL, Association for Computational Linguistics; 2020. p. 5094–107.
44. Zhu C, Hinthorn W, Xu R, Zeng Q, Zeng M, Huang X, Jiang M. Enhancing factual consistency of abstractive summarization. In: NAACL-HLT, Association for Computational Linguistics, Virtual 2021. p. 718–733.
45. Ji X, Zhao W. SKGSUM: abstractive document summarization with semantic knowledge graphs. In: IJCNN. Shenzhen: IEEE; 2021. p. 1–8.
46. Angeli G, Premkumar MJJ, Manning CD. Leveraging linguistic structure for open domain information extraction. In: ACL (1). Beijing: The Association for Computer Linguistics; 2015. p. 344–54.
47. Manning CD, Surdeanu M, Bauer J, Finkel JR, et al. The stanford corenlp natural language processing toolkit. In: ACL (System Demonstrations). Baltimore: The Association for Computer Linguistics; 2014. p. 55–60.
48. Kim J, Ohta T, Pyysalo S, Kano Y, et al. Overview of bionlp'09 shared task on event extraction. In: BioNLP@HLT-NAACL (Shared Task). Boulder: Association for Computational Linguistics; 2009. p. 1–9.
49. Kim J, Pyysalo S, Ohta T, Bossy R, Nguyen NLT, Tsujii J. Overview of bionlp shared task 2011. In: BioNLP@ACL (Shared Task). Portland: Association for Computational Linguistics; 2011. p. 1–6.
50. Nédellec C, Bossy R, Kim J, Kim J, Ohta T, Pyysalo S, Zweigenbaum P. Overview of bionlp shared task 2013. In: BioNLP@ACL (Shared Task). Sofia: Association for Computational Linguistics; 2013. p. 1–7.
51. Kim J, Ohta T, Tsujii J. Corpus annotation for mining biomedical events from literature. BMC Bioinform. 2008. https://doi.org/10.1186/1471-2105-9-10.
52. Trieu H, Tran TT, Nguyen AD, Nguyen A, et al. Deepeventmine: end-to-end neural nested event extraction from biomedical texts. Bioinformatics. 2020;36(19):4910–7. https://doi.org/10.1093/bioinformatics/btaa540.
53. Beltagy I, Lo K, Cohan A. Scibert: a pretrained language model for scientific text. arXiv preprint arXiv:1903.10676 2019.
54. Frisoni G, Moro G, Carlassare G, Carbonaro A. Unsupervised event graph representation and similarity learning on biomedical literature. Sensors. 2022;22(1):3. https://doi.org/10.3390/s22010003.
55. Devlin J, Chang M, Lee K, Toutanova K. BERT: pre-training of deep bidirectional transformers for language understanding. In: NAACL-HLT (1). Minneapolis: Association for Computational Linguistics; 2019. p. 4171–4186.

56. Liu Y, Ott M, Goyal N, Du J, Joshi M, Chen D, Levy O, Lewis M, Zettlemoyer L, Stoyanov V. Roberta: a robustly optimized BERT pretraining approach. CoRR. arXiv:1907.11692 2019.
57. Chen J, Chen H. Edge-featured graph attention network. arXiv preprint arXiv:2101.07671 2021.
58. Schlichtkrull M, Kipf TN, Bloem P, Berg Rvd, Titov I, Welling M. Modeling relational data with graph convolutional networks. In: European semantic web conference. Springer; 2018. p. 593–607
59. Levi FW. Finite geometrical systems. 1942.
60. Beck D, Haffari G, Cohn T. Graph-to-sequence learning using gated graph neural networks. arXiv preprint arXiv:1806.09835 2018.
61. Koncel-Kedziorski R, Bekal D, Luan Y, Lapata M, et al. Text generation from knowledge graphs with graph transformers. arXiv preprint arXiv:1904.02342 2019.
62. Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate. In: Bengio Y, LeCun Y, editors. 3rd International conference on learning representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings. arXiv:1409.0473 2015.
63. Fey M, Lenssen JE. Fast graph representation learning with pytorch geometric. CoRR. arXiv:1903.02428 2019.
64. Pyysalo S, Ohta T, Miwa M, Cho H, et al. Event extraction across multiple levels of biological organization. Bioinformatics. 2012;28(18):575–81.
65. See A, Liu PJ, Manning CD. Get to the point: Summarization with pointer-generator networks. In: ACL (1). Vancouver: Association for Computational Linguistics; 2017 p. 1073–083.
66. Lin C-Y. ROUGE: a package for automatic evaluation of summaries. In: Text summarization branches out. Barcelona: Association for Computational Linguistics; 2004. p. 74–81. https://aclanthology.org/W04-1013
67. Kincaid JP, Fishburne RP, Rogers RL, Chissom BS. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel. 1975.
68. Gunning R.e.a. Technique of clear writing. 1952
69. Coleman M, Liau TL. A computer readability formula designed for machine scoring. J Appl Psychol. 1975;60:283–4.
70. Frisoni G, Carbonaro A, Moro G, Zammarchi A, Avagnano M. NLG-metricverse: an end-to-end library for evaluating natural language generation. In: Proceedings of the 29th international conference on computational linguistics, international committee on computational linguistics. Gyeongju; 2022. p. 3465–479. https://aclanthology.org/2022.coling-1.306
71. Yuan H, Yuan Z, Gan R, Zhang J, Xie Y, Yu S: Biobart: pretraining and evaluation of a biomedical generative language model. In: BioNLP@ACL. Dublin: Association for Computational Linguistics; 2022. p. 97–109.
72. Schmidt V, Goyal K, Joshi A, Feld B, et al. CodeCarbon: estimate and track carbon emissions from machine learning computing. 2021. https://doi.org/10.5281/zenodo.4658424.
73. Moro G, Ragazzi L, Valgimigli L. Carburacy: summarization models tuning and comparison in eco-sustainable regimes with a novel carbon-aware accuracy. In: Thirty-seventh AAAI conference on artificial intelligence. AAAI 2023. Washington, DC: AAAI Press; 2023. p. 1–9.
74. Frisoni G, Italiani P, Salvatori S, Moro G. Cogito ergo summ: abstractive summarization of biomedical papers via semantic parsing graphs and consistency rewards. In: AAAI. AAAI Press; 2023. p. 1–9.
75. Frisoni G, Moro G, Balzani L. Text-to-text extraction and verbalization of biomedical event graphs. In: Proceedings of the 29th international conference on computational linguistics, international committee on computational linguistics, Gyeongju, Republic of Korea; 2022. p. 2692–710. https://aclanthology.org/2022.coling-1.238.
76. Moro G, Valgimigli L. Efficient self-supervised metric information retrieval: a bibliography based method applied to COVID literature. Sensors. 2021. https://doi.org/10.3390/s21196430.
77. Moro G, Salvatori S. Deep vision-language model for efficient multi-modal similarity search in fashion retrieval. In: SISAP 2022, Bologna, Italy, October 5–7, 2022, Proceedings. Lecture notes in computer science, vol. 13590. Springer; 2022. p. 40–53. https://doi.org/10.1007/978-3-031-17849-8_4
78. Moro G, Salvatori S, Frisoni G. Efficient text-image semantic search: a multi-modal vision-language approach for fashion retrieval. Neurocomputing. 2023. https://doi.org/10.1016/j.neucom.2023.03.057.
79. Domeniconi G, Masseroli M, Moro G, Pinoli P. Discovering new gene functionalities from random perturbations of known gene ontological annotations. INSTICC Press, Rome; 2014. p. 107–16. https://doi.org/10.5220/0005087801070116. https://www.scopus.com/inward/record.uri?eid=2-s2.0-84909957332 &doi=10.5220%2f0005087801070116 &partnerID=40 &md5=d46ef212e92f6a5b1c3d3769ca8a0564
80. Moro G, Masseroli M. Gene function finding through cross-organism ensemble learning. BioData Min. 2021;14(1):14. https://doi.org/10.1186/s13040-021-00239-w.
81. Chen Y, Wu L, Zaki MJ. Iterative deep graph learning for graph neural networks: Better and robust node embeddings. In: NeurIPS. 2020.
82. Domeniconi G, Moro G, Pasolini R, Sartori C. A comparison of term weighting schemes for text classification and sentiment analysis with a supervised variant of tf.idf. In: DATA (Revised Selected Papers), vol. 584. Cham: Springer; 2015. p. 39–58. https://doi.org/10.1007/978-3-319-30162-4_4. https://www.scopus.com/inward/record.uri?eid=2-s2.0-84961127206 &doi=10.1007%2f978-3-319-30162-4_4 &partnerID=40 &md5=81e9a8dc2045e1186bf840b7e43e3118
83. Domeniconi G, Moro G, Pasolini R, Sartori C. Iterative refining of category profiles for nearest centroid cross-domain text classification. In: IC3K 2014, Rome, Italy, October 21–24, 2014, Revised Selected Papers, vol. 553. Springer, Rome; 2014. p. 50–67. https://doi.org/10.1007/978-3-319-25840-9_4
84. Moro G, Pagliarani A, Pasolini R, Sartori C. Cross-domain and in-domain sentiment analysis with memory-based deep neural networks. In: IC3K 2018. Seville: SciTePress; 2018. p. 127–38. https://doi.org/10.5220/0007239101270138. https://www.scopus.com/inward/record.uri?eid=2-s2.0-85059000370 &doi=10.5220%2f0007239101270138 &partnerID=40 &md5=257a04cbdf98a4d75275d39563b0aa17
85. Ferrari I, Frisoni G, Italiani P, Moro G, Sartori C. Comprehensive analysis of knowledge graph embedding techniques benchmarked on link prediction. Electronics. 2022. https://doi.org/10.3390/electronics11233866.
86. Frisoni G, Mizutani M, Moro G, Valgimigli L. BioReader: a retrieval-enhanced text-to-text transformer for biomedical literature. In: Proceedings of the 2022 conference on empirical methods in natural language processing. Abu Dhabi: Association for Computational Linguistics. 2022. p. 5770–793. https://aclanthology.org/2022.emnlp-main.390
87. Pyysalo S, Ohta T, Ananiadou S. Overview of the cancer genetics (cg) task of bionlp shared task 2013. In: Proceedings of the BioNLP Shared Task 2013 Workshop, 2013. p. 58–66.
88. Kim J-D, Wang Y, Yasunori Y. The Genia event extraction shared task, 2013 edition—overview. In: Proceedings of the BioNLP Shared Task 2013 Workshop. Sofia: Association for Computational Linguistics; 2013. p. 8–15. https://aclanthology.org/W13-2002

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

Giacomo Frisoni[1] · Paolo Italiani[1] · Gianluca Moro[1] · Ilaria Bartolini[1] · Marco Antonio Boschetti[2] · Antonella Carbonaro[1]

✉ Giacomo Frisoni
giacomo.frisoni@unibo.it

Paolo Italiani
paolo.italiani@unibo.it

Gianluca Moro
gianluca.moro@unibo.it

Ilaria Bartolini
ilaria.bartolini@unibo.it

Marco Antonio Boschetti
marco.boschetti@unibo.it

Antonella Carbonaro
antonella.carbonaro@unibo.it

[1] Department of Computer Science and Engineering (DISI), University of Bologna, Via dell’Università 50, 47522 Cesena, Italy

[2] Department of Mathematics (DM), University of Bologna, Piazza di Porta S. Donato 5, 40126 Bologna, Italy