Entity-Driven Fact-Aware Abstractive Summarization of Biomedical Literature

Amanuel Alambo
*Wright State University - Main Campus*

Tanvi Banerjee
*Wright State University - Main Campus*, tanvi.banerjee@wright.edu

Krishnaprasad Thirunarayan
*Wright State University - Main Campus*, t.k.prasad@wright.edu

Michael Raymer
*Wright State University - Main Campus*, michael.raymer@wright.edu

Follow this and additional works at: [https://corescholar.libraries.wright.edu/cse](https://corescholar.libraries.wright.edu/cse)

Part of the Computer Sciences Commons, and the Engineering Commons

**Repository Citation**
[https://corescholar.libraries.wright.edu/cse/628](https://corescholar.libraries.wright.edu/cse/628)

This Article is brought to you for free and open access by Wright State University's CORE Scholar. It has been accepted for inclusion in Computer Science and Engineering Faculty Publications by an authorized administrator of CORE Scholar. For more information, please contact library-corescholar@wright.edu.
Abstract—As part of the large number of scientific articles being published every year, the publication rate of biomedical literature has been increasing. Consequently, there has been considerable effort to harness and summarize the massive amount of biomedical research articles. While transformer-based encoder-decoder models in a vanilla source document-to-summary setting have been extensively studied for abstractive summarization in different domains, their major limitations continue to be entity hallucination (a phenomenon where generated summaries constitute entities not related to or present in source article(s)) and factual inconsistency. This problem is exacerbated in a biomedical setting where named entities and their semantics (which can be captured through a knowledge base) constitute the essence of an article. The use of named entities and facts mined from background knowledge bases pertaining to the named entities to guide abstractive summarization has not been studied in biomedical article summarization literature. In this paper, we propose an entity-driven fact-aware framework for training end-to-end transformer-based encoder-decoder models for abstractive summarization of biomedical articles. We call the proposed approach, whose building block is a transformer-based model, EFAS, Entity-driven Fact-aware Abstractive Summarization. We conduct a set of experiments using five state-of-the-art transformer-based encoder-decoder models (two of which are specifically designed for long document summarization) and demonstrate that injecting knowledge into the training/inference phase of these models enables the models to achieve significantly better performance than the standard source document-to-summary setting in terms of entity-level factual accuracy, N-gram novelty, and semantic equivalence while performing comparably on ROUGE metrics. The proposed approach is evaluated on ICD-11-Summ-1000, a dataset we build for abstractive summarization of biomedical literature, and PubMed-50k, a segment of a large-scale benchmark dataset for abstractive summarization of biomedical literature. The contributions of this study are summarized as follows:

1. We introduce an approach for named entity-driven fact retrieval from biomedical knowledge bases using dense vector representations.
2. We propose a framework based on transformer-based encoder-decoder models for knowledge-guided large-scale abstractive summarization of biomedical literature.
3. We conduct extensive experiments and ablation studies to assess the efficacy of the proposed approach and show that injecting named entities and entity-aware facts mined from biomedical knowledge bases into abstractive summarization models can boost their performance in terms of entity-level factual consistency and N-gram novelty.

I. INTRODUCTION

Neural abstractive summarization is well explored for summarization of news articles [1]–[3], and scientific articles [6]–[8]. While there are some efforts to apply neural abstractive summarization techniques for biomedical literature [9], the use of named entities and background knowledge bases to guide biomedical abstractive summary generation has not been explored. On the other hand, named entity recognition/understanding in biomedical literature has been extensively studied such that named entities are known to harbor significant semantics about a biomedical article [10]–[13]. Further, linking named entities to concept definitions in background domain-specific knowledge bases boost semantic understanding of a biomedical article by improving comprehensiveness and contextualization as investigated in [14]–[16]. The use of named entities to guide abstractive summarization of news articles has been explored in recent studies by [17]–[20]. However, these studies do not leverage named entity-driven facts from background knowledge bases to guide abstractive summary generation. Inspired by recent advances in transformer-based encoder-decoder models [21]–[24] and knowledge and retrieval augmented natural language generation [25]–[30], we propose a technique for retrieval of entity-driven facts from biomedical knowledge bases and leveraging the retrieved facts as contextual signals for abstractive summarization of biomedical literature. Our proposed framework consists of two major components: 1) entity-driven knowledge retriever; and 2) knowledge-guided abstractive summarizer. The entity-driven knowledge retriever extracts facts from UMLS [31], ICD-10 [32], and SNOMED-CT [33] based on named entities in a biomedical article while the knowledge-guided abstractive summarizer is trained to generate summaries by attending to the source article to be summarized, the chain of named entities in the source article, and the retrieved facts from the knowledge bases.

We conduct experiments on two datasets: ICD-11-Summ-1000, and PubMed-50k. For curating ICD-11-Summ-1000, we conduct entity-driven clustering of PubMed abstracts collected per ICD-11 [34] chapter followed by entity-aware content selection to build an entity-aware pseudo extractive document. The pseudo extractive document is passed as an input source article to generate the abstractive summary during inference. The contributions of this study are summarized as follows:

• We introduce an approach for named entity-driven fact retrieval from biomedical knowledge bases using dense vector representations.
• We propose a framework based on transformer-based encoder-decoder models for knowledge-guided large-scale abstractive summarization of biomedical literature.
• We conduct extensive experiments and ablation studies to assess the efficacy of the proposed approach and show that injecting named entities and entity-aware facts mined from biomedical knowledge bases into abstractive summarization models can boost their performance in terms of entity-level factual consistency and N-gram novelty.

We propose a technique for retrieval of entity-driven facts from biomedical knowledge bases and leveraging the retrieved facts as contextual signals for abstractive summarization of biomedical literature. Our proposed framework consists of two major components: 1) entity-driven knowledge retriever; and 2) knowledge-guided abstractive summarizer. The entity-driven knowledge retriever extracts facts from UMLS [31], ICD-10 [32], and SNOMED-CT [33] based on named entities in a biomedical article while the knowledge-guided abstractive summarizer is trained to generate summaries by attending to the source article to be summarized, the chain of named entities in the source article, and the retrieved facts from the knowledge bases.

Index Terms—Transformers, Named Entity Recognition, Knowledge Bases, Abstractive Summarization, ICD-11, Knowledge Retrieval, Knowledge-enhanced Natural Language Generation

Amanuel Alambo∗, Tanvi Banerjee∗, Krishnaprasad Thirunarayan∗, Michael Raymer∗
∗Wright State University
{alambo.2, tanvi.banerjee, t.k.prasad, michael.raymer}@wright.edu
measured using BERTScore [39].

- We develop ICD-11-Summ-1000, a dataset consisting of clusters of biomedical articles collected from PubMed for the special groups chapters in the ICD-11 catalog[4] and the abstractive summaries generated using different variations of our experimental design. Further, we plan to share our code, and data with other researchers[5].

II. RELATED WORK

While abstractive summarization is well studied for summarization of news articles with success attributed to the availability of a massive amount of training data, their applicability to scholarly articles, particularly, in the biomedical domain is limited. Further, although named entities have been extensively studied to convey the semantics of an article (news, scientific, social media) and the saliency of individual sentences [18] within an article, they have not been widely used as part of modeling abstractive summarization. [17] performed entity-aware single-document abstractive summarization using reinforcement learning for training. Their pipeline-based approach consists of an entity-aware content selection module and abstract generation module. They evaluate their approach on the CNN/Daily Mail and NYT corpora. [18] perform entity-driven multi-document abstractive summarization of news articles (WikiSum, and Multi-News) using an encoder-decoder framework augmented with Graph Attention Network (GAT). [40] proposed EntityRank, an extension of the LexRank [41] graph-based algorithm, for entity-supported summarization of biomedical abstracts.

There have been a few recent efforts towards knowledge/fact-aware abstractive summarization in different domains. [42] introduced a Fact-aware Abstractive Summarization model called FaSum for improving the factual consistency of summaries in the domain of news articles. However, their approach does not leverage named entities for fact retrieval. [20] extended a transformer-based abstractive summarization model using entities disambiguated and linked to Wikidata knowledge graph and attending to the entities for summarization of news articles. Their approach, however, does not perform named-entity based fact retrieval from the knowledge base constrained by the article to be summarized and the named entities. [43] developed an unsupervised pipeline-based approach for knowledge-infused abstractive summarization for condensing patient-to-clinician diagnostic interviews based on Multi-Sentence Compression [44] and Integer Linear Programming [45]. Nevertheless, their approach uses domain-specific lexicons as knowledge source for filtering irrelevant utterances and for retrofitting language models [46] and, does not leverage named entities or facts as part of an end-to-end training of models. [47] proposed Biomed-Summarizer, a framework for extractive summarization of biomedical literature in a multi-document setting and evaluated on PubMed abstracts. [6] built a model for abstractive summarization of long documents using a discourse-aware encoder-decoder framework and experimented on two large scale datasets including research articles collected from PubMed. To address the challenge associated with the scarcity of large-scale training data in the biomedical domain, [48] released MS2 (Multi-Document Summarization of Medical Studies). They experimented with BART [22] for abstractive summary generation on the dataset they introduced in a traditional multi-doc-to-summary setting.

Though all the aforementioned studies conduct abstractive summarization of biomedical literature or the use of facts mined from knowledge bases for a different domain, they follow the well-established paradigm of source-document vs summary pairing during training/inference of models. Our approach is different in that we augment the state-of-the-art abstractive summarization models with additional contextual signals during training/inference and apply them to the biomedical domain.

III. DATA PREPARATION

In addition to the benchmark PubMed-50k [6] dataset which we use to train our models, we curate ICD-11-Summ-1000 dataset using a data preparation pipeline which follows two stages: 1) ICD-11 disease lexicon curation for querying PubMed for abstracts; and 2) Entity-aware pseudo-document generation for a collection of semantically related PubMed abstracts in an ICD-11 chapter. Thus, for querying PubMed for abstracts for an ICD-11 chapter, we first query the biomedical knowledge bases for “disease” keywords using the names of each ICD-11 chapter and build a lexicon of diseases corresponding to each chapter. Figure-1 shows what a lexicon build-up for an ICD-11 chapter looks like. Once the disease related keywords are identified for an ICD-11 chapter, we use these keywords to query PubMed via the Bio Entrez parser[2] to capture the first 1000 abstracts (PMIDs) spanning a period of last 90 days from the moment we initiated the query. We do this for each of the eight special groups ICD-11 chapters.

---

2. https://github.com/AmanuelF/biomed_abstractive_summarization
abstracts within an ICD-11 chapter, we first conduct clustering of the named entities using agglomerative clustering as used in [13]. We use BioBERT [50] for named entity representation followed by agglomerative clustering. Once the named entities pertaining to an ICD-11 chapter are clustered into different bins, our next task is to cluster the PubMed abstracts into a bin based on how the named entities within the abstracts are related to the entities within a cluster. We use cosine similarity between named entities identified in a PubMed abstract and entities characterizing a cluster to determine an entity-aware cluster the abstract belongs to. Next, for each cluster, we perform named entity-aware salient content selection to produce an extractive pseudo-document for each cluster. This paradigm of reducing a multi-document corpus (i.e., a cluster consisting of PubMed abstracts grouped based on entity-relatedness) into an extractive pseudo-document is explored for different tasks in previous studies [51]–[53]. As part of the NER task, we use coreference resolution [24], [25] after named entities are extracted using SciSpacy to cluster the biomedical named entities and their coreferenced mentions spanning the multiple abstracts within an ICD-11 chapter.

For all tasks throughout this paper involving initializing of networks or for representation learning, we use BioBERT [50].

IV. PROPOSED METHOD

Our proposed approach is a two-stage framework consisting of 1) an entity-driven knowledge retriever, and 2) a knowledge-guided abstractive summarizer. In this section, we discuss both modules in detail.

A. Entity-driven Knowledge Retriever

For each extractive pseudo-document generated in the data preparation stage for ICD-11 or input article for PubMed-50k designated by \( D \), we identify the named entities in the input document. The identified named entities are then used to retrieve facts from biomedical knowledge bases (UMLS, ICD-10, and SNOMED-CT). We use PyMedTermino [57] to work with the entire dump of UMLS [31] available at [31]. For \( m \) named entities (and their coreferenced mentions), we have a set of pairs of entities \( \{(e_i, e_j) \mid 0 \leq i < j < m \} \) extracted from \( D \), where each pair \( (e_i, e_j) \) is used to query for \( c \) candidate facts \( F_1, F_2, F_3, \ldots, F_c \) denoted collectively by \( F_D \) from the background knowledge bases \( K \) using full text search. The complete set of facts retrieved for all pairs of named entities in source document \( D \) is denoted by \( F_D \).

The reason we use a pair of named entities to perform lexical retrieval from \( K \) is to capture the relationship between a pair of named entities as it appears in a knowledge base since relationships among named entities harbor semantics in addition to the entities themselves and help with disambiguating relevant facts from irrelevant facts. After the candidate facts \( F_D \) are retrieved from the knowledge bases \( K \), we embed the candidate facts using BioBERT. Then, we perform efficient vector similarity search using Maximum Inner Product Search (MIPS) [58] implemented in the FAISS library [59] to query for the top-\( k \) facts among the candidate facts (\( F_D \)) using the input document \( D \) as the query. Formally, we define the similarity between fact \( F_i \in F_D \) and document \( D \) as

\[
\text{sim}(F_i, D) = \overrightarrow{V}(F_i)^T \overrightarrow{V}(D)
\]

where \( \overrightarrow{V}(F_i) \) - Vector representation of Fact \( F_i \); \( \overrightarrow{V}(D) \) - Vector representation of document \( D \).

Thus, after the knowledge retrieval task, we have 1) the input document \( D \) which is obtained during the data preparation phase for ICD-11 and readily available for PubMed-50k; 2) the named entity chain (i.e., chain of named entities extracted from the pseudo-doc) \( E \) [19]; and 3) top-\( k \) facts \( F_1, F_2, F_3, \ldots, F_K \) retrieved from the background knowledge bases collectively represented as \( F_K \subseteq F_D \). We set the value of \( K \) to 3 following the study by [30]. We experiment with different values of \( K \) as detailed in the ablation studies section. The combination of these contextual signals will be used to guide the summarization model at training/inference time. The rationale for using maximum inner product search for knowledge retrieval

\[\text{https://bit.ly/3E0zIl}\]

\[\text{https://github.com/facebookresearch/faiss}\]
is inspired by the works of [26]-[28], [59], [60], albeit they used it mainly for open domain question answering [61], [62], [30] use a similar approach for exemplar retrieval in their RetrievalSum model which is based on contrastive learning [63] using a Siamese network [64] to learn representations for an input document and the exemplars and guide their summary generation. Our problem of retrieving the most relevant facts from the background KB, however, is framed as a dense passage retrieval problem. Named entities from the input document are extracted using the SciSpacy NER model trained on the BCCSDR corpus [49]. Table-I shows sample facts, as they appear and retrieved from UMLS KB for an input article with a given pair of named entities identified.

<table>
<thead>
<tr>
<th>Named Entity Pair</th>
<th>Entity-driven Facts from UMLS KB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type 1 diabetes</td>
<td>Hemoglobin A1C in type 1 diabetes.</td>
</tr>
<tr>
<td>Type 2 diabetes</td>
<td>Hemoglobin A1C in type 2 diabetes.</td>
</tr>
<tr>
<td>Hypertension</td>
<td>Hypertension in chronic kidney disease due to type 1 diabetes mellitus.</td>
</tr>
<tr>
<td>Cardiovascular disease</td>
<td>Hypertension and heart failure due to type 2 diabetes mellitus.</td>
</tr>
</tbody>
</table>

TABLE I: Pairs of named entities and sample facts mined from UMLS for each pair.

B. Knowledge-guided Abstractive Summarizer

The backbone component of our knowledge-guided abstractive summarizer, which is a transformer encoder-decoder model, is based on the work by [65]. Figure-3 shows the proposed end-to-end model architecture. We use this architectural setup for all the models we experiment with. We designate a model augmented with one of the knowledge signals as model-ENAS. We train the models on the 50k samples obtained from PubMed abstractive scientific summarization dataset [6] using different combinations of signals (with and without named entities and facts).

The top-k facts retrieved by the biomedical knowledge retriever, corresponding to each pair of named entities in an input extractive pseudo-doc or input article, are separated by a special token [SEP]. The input article is passed as one input document prepended with [CLS] and appended with [SEP] token. The named entity chain is passed as one segment prepended with [CLS] and appended with [SEP] token. There have been different approaches to combining different signals such as concatenating the different pieces to prime the generation component such as the one proposed in Fusion-in Decoder [59] and [60]. The top-k retrieved facts are initialized using BioBERT and the concatenated encoding is then passed through a sequence of transformer layers to be projected onto a 768- dimension vector to later be attended to by the autoregressive decoder. Similarly, the named entity chain is initialized with BioBERT and passed through a sequence of transformer encoders. Each transformer encoder layer is composed of self-attention and feed forward sub-layers. At training time, a batch of input-output pairings is passed to the encoder and decoder respectively in the form \(\langle x, y \rangle\). The encoder undergoes the following transformations to the input sequence \(x\) which in this formalism is used to represent the first hidden layer \(h^0\) of the stacked sequence of \(l\) transformer encoder layers.

\[
\begin{align*}
    h^0_l & := \text{LayerNorm}(h^{l-1}_S + \text{MMHAtt}(h^{l-1}_S)) \\
    h^l_0 & := \text{LayerNorm}(h^l_S + \text{FFN}(h^l_S))
\end{align*}
\]

V. MODEL TRAINING

All models are trained with a cross-entropy loss using back-propagation:

\[
L_\theta = -\frac{1}{n} \sum_{k=1}^{n} P(t_k|x_k, X, \mathcal{E}, F_K; \theta)
\]  

Where \(X\) - the input sequence to be summarized; \(\mathcal{E}\) - the named entities chain in the input sequence \(X\); and \(F_K\) - top-k facts extracted from biomedical KB; \(\theta\) - model parameters.

We train each model using cross entropy loss to generate the ground truth summaries for the PubMed-50k dataset. We use the following hyperparameters setting: number of epochs to 5, fixed learning rate of 5e-5 with adam optimizer [67], batch size to 8, beam size of 5 with a length penalty [68] \(\alpha\) between the range of 0.6 and 1 [5] at inference time; to deal with long-document summarization using the traditional transformer encoder-decoder models, we split the source article into
chunks of a maximum of 512 tokens and independently encode each chunk, after which we concatenate and project back to 768 dimension using a linear layer. The approach of splitting the long input sequence into smaller chunks of 512 tokens and then embedding independently is motivated by the recent work by [48]. For Longformer-Encoder-Decoder (LED) [69] and BigBird [70], however, we set the maximum length of the input sequence to 8192 tokens since they can deal with long input sequences without having to truncate; maximum output sequence length is set to 210 tokens following the experiments by [6]. To mitigate redundancy in the generated summaries, we enable trigram blocking [71] during inference. For each backbone model, we use its base variant with 12 encoder and 12 decoder layers. The train/validation/test sizes for PubMed-50k is 50,000/5,000/5,000 and each model is trained using early stopping. A checkpoint of the model that performs the best (in terms of validation loss) on the validation set across different epochs is saved to the file system. All models are built and trained using PyTorch on NVIDIA Tesla T4 GPU. We perform model training experiments with different input guidance settings: input document only, input document + named entities chain, input document + named entities chain + knowledge facts. For our base summarization model, we experiment with five transformer-based encoder-decoder models and show that our entity-driven knowledge-aware approach enables us to achieve the best performance in entity-level factual consistency, N-gram novelty, and semantic equivalence while performing comparably on the commonly used ROUGE metrics. At inference time, we experiment with two settings (w/o named entities, and w/ named entities).

VI. EXPERIMENTS AND RESULTS

While all models are trained using the PubMed-50k corpus, they are evaluated using a hold-out test set from the original PubMed dataset as well as the ICD-11-Summ-1000 corpus we curate. The experimental results are shown in Table-II through Table-IX. Results of evaluation w.r.t source articles reported are average results for both the ICD-11-Summ-1000 and PubMed corpora since the ICD-11 pseudo-extractive documents do not have a ground truth summary. For lexical (ROUGE) evaluation, we report ROUGE F1 scores [72]. Similarly, here for evaluation conducted w.r.t source articles, the results reported are average results across the PubMed and ICD-11-Summ-1000 corpora. Entity-level factual accuracy [37] is measured in terms of precision, and recall w.r.t ground truth summary (for PubMed), and w.r.t source articles (for both PubMed and ICD-11-Summ-1000). Entity-level precision and recall w.r.t ground truth summaries are denoted with precision-target and recall-target; similarly, entity-level precision, and recall w.r.t the source article are designated with precision-source, and recall-source, respectively. The F1 score is the harmonic mean of the precision and recall for either case. For measuring semantic equivalence between generated summaries and ground truth summaries, we leverage BERTScore as proposed by [39]: specifically, we use BioBERT for representing each token in a generated summary and the ground truth summary after which we perform pairwise cosine similarity as proposed in [39]. All experimental results are reported in percentages. The average full text length of input source articles in PubMed-50k is 3,224 words and the average abstract length is 218 words, while for ICD-11-Summ-1000, the average length of an extractive pseudo-doc (i.e., input source article) is 4816 words.

VII. ABLATION STUDIES

To assess the impact of facts mined on the quality of summaries generated, we conduct an ablation study where we experiment with different values of $K$ in top-$K$ for the backbone model. Figure-4 shows results of ablation to assess precision-source and recall-target. Since we want to minimize entity hallucination which is measured in terms of precision-source and want to maximize the number of entities in the ground truth summary that are retrieved in the generated summary as measured by recall-target, we report the impact of different values of $K$ for these two metrics. As shown in the two plots, precision-source and recall-target consistently improve as we increase $K$.

<table>
<thead>
<tr>
<th>Backbone Model</th>
<th>Training Config (K=3)</th>
<th>Recalling</th>
<th>Precision-source</th>
<th>Recall-source</th>
<th>Precision-target</th>
<th>Recall-target</th>
</tr>
</thead>
<tbody>
<tr>
<td>T5</td>
<td>w/o named entities (Ours)</td>
<td>26.315</td>
<td>12.285</td>
<td>29.955</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BART</td>
<td>w/o named entities (Ours)</td>
<td>28.667</td>
<td>13.285</td>
<td>31.545</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pegasus</td>
<td>w/o named entities (Ours)</td>
<td>25.851</td>
<td>11.247</td>
<td>26.859</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BigBird</td>
<td>w/o named entities (Baseline)</td>
<td>30.872</td>
<td>12.915</td>
<td>26.503</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LED</td>
<td>w/o named entities (Ours)</td>
<td>35.426</td>
<td>13.901</td>
<td>32.533</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

TABLE II: Lexical (ROUGE) Evaluation w.r.t Ground Truth Summary (vanilla input @ inference time). The input in this experimental setting is the raw input article to be summarized (i.e., w/o named entity chain)
### TABLE V: Lexical (ROUGE) Evaluation w.r.t Ground Truth Summary (input article + named entity chain @ inference time): i.e., the input in this experimental setting is the raw input article to be summarized with the named entities (i.e., w/ named entity chain).

<table>
<thead>
<tr>
<th>Backbone Model</th>
<th>Training Config (v3)</th>
<th>ROUGE-1</th>
<th>ROUGE-2</th>
<th>ROUGE-3</th>
</tr>
</thead>
<tbody>
<tr>
<td>T5</td>
<td>w/ named entities</td>
<td>34.927</td>
<td>30.455</td>
<td>26.827</td>
</tr>
<tr>
<td>BART</td>
<td>w/ named entities</td>
<td>36.214</td>
<td>32.758</td>
<td>29.278</td>
</tr>
<tr>
<td>Pegasus</td>
<td>w/ named entities</td>
<td>34.927</td>
<td>30.455</td>
<td>26.827</td>
</tr>
<tr>
<td>BigBird</td>
<td>w/ named entities</td>
<td>34.927</td>
<td>30.455</td>
<td>26.827</td>
</tr>
<tr>
<td>LED</td>
<td>w/ named entities</td>
<td>34.927</td>
<td>30.455</td>
<td>26.827</td>
</tr>
</tbody>
</table>

### TABLE VI: Entity-level Factual Consistency w.r.t Ground Truth Summary. The input in this experimental setting is the raw input article to be summarized with the named entities (i.e., w/ named entity chain) @ inference time.

<table>
<thead>
<tr>
<th>Backbone Model</th>
<th>Training Config (v3)</th>
<th>Factual Consistency</th>
</tr>
</thead>
<tbody>
<tr>
<td>T5</td>
<td>w/ named entities</td>
<td>34.927</td>
</tr>
<tr>
<td>BART</td>
<td>w/ named entities</td>
<td>36.214</td>
</tr>
<tr>
<td>Pegasus</td>
<td>w/ named entities</td>
<td>34.927</td>
</tr>
<tr>
<td>BigBird</td>
<td>w/ named entities</td>
<td>34.927</td>
</tr>
<tr>
<td>LED</td>
<td>w/ named entities</td>
<td>34.927</td>
</tr>
</tbody>
</table>

### TABLE VII: Entity-level Factual Consistency w.r.t input source article (input article + named entity chain @ inference time): i.e., the input in this experimental setting is the raw input article to be summarized with the named entities (i.e., w/ named entity chain).

<table>
<thead>
<tr>
<th>Backbone Model</th>
<th>Training Config (v3)</th>
<th>Factual Consistency</th>
</tr>
</thead>
<tbody>
<tr>
<td>T5</td>
<td>w/ named entities</td>
<td>34.927</td>
</tr>
<tr>
<td>BART</td>
<td>w/ named entities</td>
<td>36.214</td>
</tr>
<tr>
<td>Pegasus</td>
<td>w/ named entities</td>
<td>34.927</td>
</tr>
<tr>
<td>BigBird</td>
<td>w/ named entities</td>
<td>34.927</td>
</tr>
<tr>
<td>LED</td>
<td>w/ named entities</td>
<td>34.927</td>
</tr>
</tbody>
</table>

### TABLE VIII: N-gram Novelty w.r.t source articles w/o and w/ named entity chain during inference.

<table>
<thead>
<tr>
<th>Backbone Model</th>
<th>Training Config (v3)</th>
<th>N-gram Novelty</th>
</tr>
</thead>
<tbody>
<tr>
<td>T5</td>
<td>w/ named entities</td>
<td>34.927</td>
</tr>
<tr>
<td>BART</td>
<td>w/ named entities</td>
<td>36.214</td>
</tr>
<tr>
<td>Pegasus</td>
<td>w/ named entities</td>
<td>34.927</td>
</tr>
<tr>
<td>BigBird</td>
<td>w/ named entities</td>
<td>34.927</td>
</tr>
<tr>
<td>LED</td>
<td>w/ named entities</td>
<td>34.927</td>
</tr>
</tbody>
</table>

### TABLE IX: Semantic Equivalence (BioBERTScore) w.r.t ground truth summaries w/o and w/ named entity chain during inference. Since we are using BioBERT for representation learning, we refer to the metric as BioBERTScore, a variant of BERTScore.

<table>
<thead>
<tr>
<th>Backbone Model</th>
<th>Training Config (v3)</th>
<th>BioBERTScore</th>
</tr>
</thead>
<tbody>
<tr>
<td>T5</td>
<td>w/ named entities</td>
<td>34.927</td>
</tr>
<tr>
<td>BART</td>
<td>w/ named entities</td>
<td>36.214</td>
</tr>
<tr>
<td>Pegasus</td>
<td>w/ named entities</td>
<td>34.927</td>
</tr>
<tr>
<td>BigBird</td>
<td>w/ named entities</td>
<td>34.927</td>
</tr>
<tr>
<td>LED</td>
<td>w/ named entities</td>
<td>34.927</td>
</tr>
</tbody>
</table>

VIII. DISCUSSION OF RESULTS

From the results reported in the previous section, we generally see entity-level factual consistency (particularly, precision-source, and recall-target) improve when a model is trained with named entities and/or facts included as an additional signal in the training with the same objective of generating the ground truth summary using cross-entropy loss. The addition of more facts further improves entity-level factual consistency as shown in Figure-4. Further, we notice N-gram novelty improves with our proposed framework for the five backbone models. Semantic equivalence generally improves when named entities and/or facts are included during training for all models. Thus, the corresponding entries for the various models and training configurations show improvement in semantic based scores. The ROUGE scores, however, drop slightly from when there is no additional context at training or inference time. The drop in ROUGE is a result of augmenting the models with facts from background knowledge bases which in turn leads to higher N-gram novelty. Thus, the proposed framework enables us to achieve better abstractive scores in terms of entity-level factual consistency, paraphrasing and semantic equivalence.

IX. CONCLUSIONS AND FUTURE WORK

In this study, we proposed a framework to integrate named entities in a source article and facts extracted from biomedical knowledge bases pertaining to the named entities in transformer-based encoder-decoder models and applied to the task of abstractive summarization of biomedical literature. Through extensive experiments, we showed the proposed approach improves the semantics of generated summaries in terms of entity-level factual consistency and semantic equivalence while generating novel words. For future steps, we plan to jointly train the knowledge-retriever and the knowledge-guided abstractive summarizer in an end-to-end fashion. While our current architecture optimizes a single cost function given different input signals, we plan to augment the existing framework using multi-objective optimization to further enhance the factual accuracy and semantic equivalence of generated summaries using different cost functions during training.

---

![Figure 4: Precision-source and Recall-target for different values of K.](image-url)
REFERENCES


