Factual Error Correction for Abstractive Summaries Using Entity Retrieval

Hwanhee Lee[1], **Cheoneum Park**[2], **Seunghyun Yoon**[3],
Trung Bui[3], **Franck Dernoncourt**[3], **Juae Kim**[2] and **Kyomin Jung**[1]
[1]Automation and Systems Research Institute, Seoul National University
[2]42dot, Hyundai Motor Group, [3]Adobe Research
{wanted1007,kjung}@snu.ac.kr, {parkce3, jju75474}@gmail.com
{syoon, bui, franck.dernoncourt}@adobe.com

Abstract

Despite the recent advancements in abstractive summarization systems leveraged from large-scale datasets and pre-trained language models, the factual correctness of the summary is still insufficient. One line of trials to mitigate this problem is to include a post-editing process that can detect and correct factual errors in the summary. In building such a system, it is strongly required that 1) the process has a high success rate and interpretability and 2) it has a fast running time. Previous approaches focus on the regeneration of the summary, resulting in low interpretability and high computing resources. In this paper, we propose an efficient factual error correction system RFEC based on entity retrieval. RFEC first retrieves the evidence sentences from the original document by comparing the sentences with the target summary to reduce the length of the text to analyze. Next, RFEC detects entity-level errors in the summaries using the evidence sentences and substitutes the wrong entities with the accurate entities from the evidence sentences. Experimental results show that our proposed error correction system shows more competitive performance than baseline methods in correcting factual errors with a much faster speed.[1]

1 Introduction

Text summarization is a task that aims to generate a short version of the text that contains the important information for the given source article. With the advances of neural text summarization systems, abstractive summarization systems (Nallapati et al., 2017) that generate novel sentences rather than extracting the snippets in the source are widely used (Lin and Ng, 2019). However, factual inconsistency between the original text and the summary is frequently observed in the abstractive summarization system (Cao et al., 2018; Zhao et al., 2020; Maynez et al., 2020). As in the example of Figure 1, many of these errors in the summaries occur at the entry-level such as person name and number. But these types of errors are sometimes trivial and can often be easily solved through simple modification like changing the wrong entities, as shown in Figure 1. For this reason, previous works (Cao et al., 2020; Zhu et al., 2021; Thorne and Vlachos, 2021) have introduced post-editing systems to alleviate these factual errors in the summary. However, all of those works adopt the seq2seq model, which requires a similar cost to the original abstractive summarization systems, as a post-editing. Therefore, using such systems based on seq2seq doubles the inference time for performing post-editing, resulting in significant inefficiency. In addition, seq2seq based post-editing model can be affected by the model's own bias to the input summary.

Article: *Singer-songwriter **David Crosby** hit a jogger with his car Sunday evening, a spokesman said.* The accident happened in Santa Ynez, California, near where Crosby lives. Crosby was driving at approximately 50 mph when he struck the jogger. The posted speed limit was 55. The jogger suffered multiple fractures, and was airlifted to a hospital in Santa Barbara, Clotworthy said.,...

System Summary with Factual Error: ***Don Clotworthy*** *hit a jogger with his car Sunday evening.* The jogger suffered multiple fractures and was airlifted to a hospital.

After Correction: ***David Crosby*** *hit a jogger with his car Sunday evening.* The jogger suffered multiple fractures and was airlifted to a hospital.

Figure 1: An example of generated summary with factual errors and the correct summary after minor modification.

To overcome this issue and develop an efficient factual corrector for summarization systems, we propose a different approach, RFEC(**R**etrieval-based **F**actual **E**rror **C**orrector) that efficiently corrects the factual errors with a much faster running time compared to seq2seq model. RFEC first retrieves the evidence sentences for the given summary for correcting and detecting errors. By doing so, we shorten the input length of the model to obtain computational efficiency. Then, RFEC examines all of the entities to determine whether each

[1] https://github.com/hwanheelee1993/RFEC

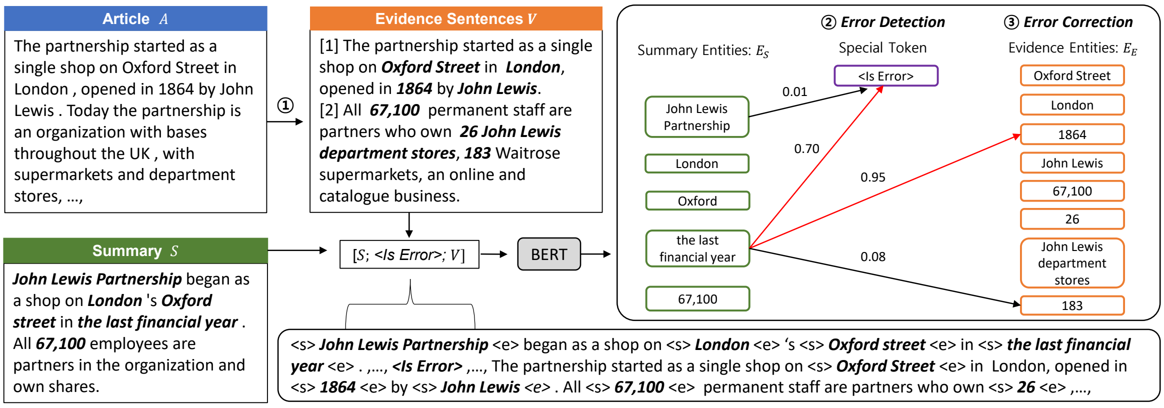
Article A
The partnership started as a single shop on Oxford Street in London , opened in 1864 by John Lewis . Today the partnership is an organization with bases throughout the UK , with supermarkets and department stores, ...,
①
Evidence Sentences V
[1] The partnership started as a single shop on Oxford Street in London, opened in 1864 by John Lewis.
[2] All 67,100 permanent staff are partners who own 26 John Lewis department stores, 183 Waitrose supermarkets, an online and catalogue business.
Summary S
John Lewis Partnership began as a shop on London 's Oxford street in the last financial year . All 67,100 employees are partners in the organization and own shares.
[S; <Is Error>; V]
BERT
<s> John Lewis Partnership <e> began as a shop on <s> London <e> 's <s> Oxford street <e> in <s> the last financial year <e> . ,..., <Is Error> ,..., The partnership started as a single shop on <s> Oxford Street <e> in London, opened in <s> 1864 <e> by <s> John Lewis <e> . All <s> 67,100 <e> permanent staff are partners who own <s> 26 <e> ,...,
② Error Detection
③ Error Correction
Summary Entities: E_S
Special Token
Evidence Entities: E_E
<Is Error>
John Lewis Partnership
London
Oxford
the last financial year
67,100
0.01
0.70
0.95
0.08
Oxford Street
London
1864
John Lewis
67,100
26
John Lewis department stores
183

Figure 2: Overall flow of our proposed RFEC. Given a summary S and an article A, we first retrieve evidence sentences V. Using S and V, we compute BERT embeddings for entities in summary E_S and evidence sentence V. If the erroneous score computed using a special token <*Is Error*> is above threshold, we regard those entity as an error and substitute it with one of the entities in the evidence sentences that obtains highest score.

entity has a factual error. If any entities have a factual error, RFEC substitutes these wrong entities with the correct entity by choosing them among the entities in the source article. Through these steps, we do not create a whole sentence as in the seq2seq model, but decide whether to fix and correct it through the retrieval, resulting in higher computational efficiency. Experiments on both synthetic and real-world benchmark datasets demonstrate that our model shows competitive performance with the baseline model with much faster running time. Also, as shown in Figure 2, RFEC has a natural form of interpretability through the visualization of the erroneous score and the scores of each candidate entity for correcting the wrong entities.

2 Related Work

With the advancement of pre-training language models such as BERT (Devlin et al., 2019) and BART (Lewis et al., 2020), abstractive summarization systems have adopted these models to use the rich information inherent in parameters. While these models improved the performance, the generated summaries are still often factually inconsistent with the source article. (Pagnoni et al., 2021).

To solve the factual inconsistency in abstractive summarization systems, FASUM (Zhu et al., 2021) adopted graph attention network (Veličković et al., 2018) for generating the correction summary. Chen et al. (2021) studied contrast candidate generation and selection by ranking approach as a model-agnostic post-processing technique to correct the extrinsic hallucinations.

Another line of mitigating factual errors is to develop a post-editing system to fix the errors. Cao et al. (2020) presented a post-editing corrector module using a BART-based auto-regressive model. The study generated a corrupted summary to train the correction system by substituting the key information, such as an entity or a number, to construct a training dataset. Thorne and Vlachos (2021) also develop a seq2seq based error correction system in the claim of FEVER dataset (Thorne et al., 2018) by correcting the words after masking some words. Different from seq2seq based previous works, we develop a faster retrieval based factual error correction system that does not generate the whole summary, only corrects the entity-level errors by substituting them with one of the entities in the article.

3 Method

3.1 Problem Formulation

For a given summary S and an article A, we aim to develop a fast retrieval-based factual error correction system that can fix the possible factual errors in S. Since factual errors frequently appear in entity-level (Goyal and Durrett, 2021), we develop a system that is specialized in correcting entity-level errors. Specifically, we define this problem as two steps, entity-level error detection and entity-level error correction as shown in Figure 2. For given n_s entities $E_S = \{es_1, es_2, ..., es_{n_s}\}$ in a summary S, we first classify whether each entity is factually consistent with the article A. If any entity e_{S_i} is factually inconsistent, the system substitutes it with one of the n_a entities in the article $E_A = \{ea_1, ea_2, ..., ea_{n_a}\}$.

3.2 Training Dataset Construction

To train a factual error correction system, we need a triple composed of an input summary S_1 that may have factual errors, an article A and a target summary S_2 that is a modified version of S_1 without factual errors. However, it is difficult to obtain S_1 that has the errors with the position annotated and the right ground truth correction of such errors. Hence we construct a synthetic training dataset by editing the reference summaries following previous works (Cao et al., 2020; Zhu et al., 2021; Kryscinski et al., 2020). We corrupt reference summaries in CNN/DM dataset (Nallapati et al., 2016) by randomly changing one of the entities with the same type of other entities in the dataset to make a corrupted summary. Finally, we construct a triple (S_1, A, S_2). Meanwhile, in the real-world dataset, a significant number of summaries are factually consistent, so we only make errors for 50% of the summaries and set $S_1 = S_2$ for the rest of the summaries in the dataset.

3.3 Evidence Sentence Retrieval

Generally, a summary does not treat all of the contents in the article but only contains some important parts of the article. Hence, in most cases, checking for errors within the summary and correcting them does not require the entire article, and using the part related to the summary is sufficient, as shown in Figure 2. Inspired by this observation, we extract some of the sentences in the article according to the similarity with the summary to increase the efficiency of the system by shortening the input length. We use ROUGE-L (Lin, 2004) score as a similarity measure to extract top-2 evidence sentences for each sentence in the summary. Then, we remove the duplicate sentences and sort them according to the order in which they appear in the article, and combine them to form $V = \{V_1, V_2, ..., V_M\}$, a set of evidence sentences for detecting and correcting errors in the summary S.

3.4 Entity Retrieval Based Factual Error Correction

Computing Embedding Using summary S and the evidence sentences V, we first extract entities E_S and E_V respectively with SpaCy[2]. And we insert special tokens <s> and <e>, before and after each extracted entity. Then we also insert an additional token *<Is Error>*, which is later used for checking the factual consistency between S and V and concatenate them to make an input for the BERT (Devlin et al., 2019). Using BERT, we obtain the contextualized embedding of each entity in S and V as follows:

$$H{=}[h_1,h_2,...,h_l]{=}BERT([S;{<}IsError{>};V]), \tag{1}$$

where l is the maximum sequence length of the input.

And we get the embedding of start token <s> for each entity as the entity embeddings $HE_V = \{h_{ev_1}, h_{ev_2}, ..., h_{ev_{n_v}}\}$ and $HE_S = \{h_{es_1}, h_{es_2}, ..., h_{es_{n_s}}\}$ for V and S respectively. We also get h_{er}, an embedding of *<Is Error>*.

Error Detection Using the computed embeddings, we compute the erroneous score for all of the entities in summary using h_{er} as follows.

$$\hat{s}_{er_i}{=}P(Err|es_i){=}\sigma(h_{es_i}^{\intercal}\ W_{dt}h_{er}{+}b_{dt}), \tag{2}$$

where $i = 1, 2, 3, ..., n_s$. The W_{dt} and b_{dt} are model parameters.

Error Correction For the entities that are factual errors, we compute the correction score between the entities and all of the entities in the evidence sentences similar to error detection as follows.

$$\hat{s}_{cr_{ij}}{=}P(Cor|es_i,ev_j){=}\sigma(h_{es_i}^{\intercal}\ W_{cr}h_{ev_j}{+}b_{cr}), \tag{3}$$

where $i = 1, 2, 3, ..., n_{s_{er}}$, $j = 1, 2, 3, ..., n_v$. $n_{s_{er}}$ is the number of errors in the summary. The W_{cr} and b_{cr} are model parameters.

Training Objective We train the model using binary cross entropy loss for both detection and correction through multi-task learning as follows.

$$L_{dt}{=}-\frac{\sum_{i=1}^{n_s}(s_{er_i}\log(\hat{s}_{er_i}){-}(1{-}s_{er_i})\log(1{-}\hat{s}_{er_i}))}{n_s} \tag{4}$$

$$L_{cr}{=}-\frac{\sum_{i=1}^{n_s}\sum_{j=1}^{n_v}(s_{cr_{ij}}\log(\hat{s}_{cr_{ij}}){-}(1{-}s_{cr_{ij}})\log(1{-}\hat{s}_{cr_{ij}}))}{n_s{\cdot}n_v} \tag{5}$$

$$L{=}L_{dt}{+}L_{cr}, \tag{6}$$

where $s_{er_i} \in \{0, 1\}$ and $s_{cr_{ij}} \in \{0, 1\}$, which are the ground truth labels for detection and correction.

[2] https://spacy.io/api/entityrecognizer

Inference For the inference stage, we do not have the label as to whether each entity is an error. Therefore, we calculate the two results sequentially, error detection and error correction, using the same BERT embeddings. For each entity, if an erroneous score is above thr_{dt}, then we let that entity be an error as shown in Figure 2. And then, we search the candidate of correction among the evidence entities HE_V, and substitute it with the entity that gets the maximum score as in Figure 2. We conduct correction only when the maximum score is higher than thr_{cr} to prevent unnatural correction caused by failure to find the appropriate entity within the candidate.

4 Experiments

To evaluate the performance of the proposed error correction system, we measure the success rate of correction for the systems by comparing the correction results with the ground-truth summaries or conducting a human evaluation for the corrected summaries as follows.

4.1 Benchmark Datasets

We evaluate our proposed factual error correction method on one synthetic testset and one real-world testset, based on CNN/DM. For the synthetic testset, we use the same method in Section 3.2 to make separate 3k samples using the test split of CNN/DM. For this dataset, we measure the success rate of the correction by comparing the corrected summary from the model with the ground truth summary. In addition to this synthetic data, we also use the FactCC-Test set (Kryscinski et al., 2020) that has labels on the 503 system-generated summaries whether they are factually consistent or not, as in (Cao et al., 2020). There are 62 inconsistent summaries, and 441 consistent summaries in this dataset. Different from the synthetic testset, FactCC-Test dataset does not provide the ground truth correction for the inconsistent summaries. Hence, we manually conduct a blind test to check the factual consistency of each summary after the correction for all of the systems as in the example of Figure 3.

4.2 Implementation Details

For our experiments, we use *bert-base-cased*[3] for RFEC. We train the model for five epochs with a learning rate of 3e-5. For baseline seq2seq model, we use *bart-base*[4] following the settings in the previous work (Cao et al., 2020) and train the model using the same dataset to correct the errors in the input summary. We search the hyperparmeters through the correction accuracy in the validation set among the five epochs. We set batch size of 32 for RFEC and 64 for BART models. We set both thr_{det} and thr_{cor} for 0.5 using the validation set. For maximum sequence length, we set 1024 for BART, 256 for BART with evidence selection, 256 for RFEC, and 512 for RFEC without evidence sentence selection. We measure the running time, including the preprocessing time of each method using a single A5000 GPU and Intel(R) Xeon(R) Silver 4210R CPU (2.40 GHz). We make the best effort to set the maximum batch size for each method using the same environment for a fair comparison.

4.3 Performance Comparison

Synthetic Dataset We present the results for the 3k synthetic testset in Table 1. We measure the correction accuracy by checking whether the corrected summary is same as the ground-truth summary for this dataset. We observe that the performance of BART is slightly better than RFEC, but our proposed retrieval-based model has a higher efficiency from the eight times faster running time. Also, we find that using only evidence sentences shows slightly less performance, but has advantages in computing speed for both systems. Especially for RFEC, it does not take much time to calculate the model output, but it costs relatively much time on preprocessing mostly on named entity recognition. And reducing the input length through the sentence selection also reduces the running time with a slight decrease in performance as shown in Table 1.

Method	Sample/min	Accuracy
Seq2seq - BART	933	90.93
- sentence selection	629	**92.20**
RFEC	**4024**	91.06
- sentence selection	1810	91.15

Table 1: Factual error correction results on test split of synthetic Test Dataset with the average running time.

FactCC-Test Dataset We present the results for the FactCC-Test dataset through the changes in factual consistency after the correction in Table 2. Compared to the results in the synthetic dataset, both seq2seq and RFEC do not correct many errors,

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

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

only 9 and 7 for the best settings in both systems among 62 errors. As in the synthetic dataset, our proposed method shows almost the same results with less running time compared to the seq2seq method. Also, we can observe that using the correction model also creates a significant number of new errors (i.e. consistent->inconsistent) especially for the seq2seq model without sentence selection.

Method	Inconsistent(62)		Consistent(441)	
	Changed	Edited	Changed	Edited
Seq2seq - BART	8	15	**2**	14
- sentence selection	**9**	23	7	78
RFEC	7	9	**2**	23
- sentence selection	6	8	3	31

Table 2: Factual error correction results on FactCC-Testset. Each column represents how many corrections each system has performed for the sample of each label, and how many labels have changed from the correction.

4.4 Qualitative Analysis

We present the representative success and failure cases of our proposed retrieval-based factual error correction system with the top-3 retrieved entities for the errors in Figure 3. For the first example, RFEC successfully corrects the error *Valerie Braham* by substituting it with *Philippe Braham* which gets a higher correction score among the entities in the evidence sentences. Also, as the object to be corrected is a person's name, we can observe that other correction candidates are also names. On the other hand, for the second example, although RFEC detects the error *Raymond*, but does not find the correction candidates whose correction score is above thr_{cr}. For this example, *Raymond* should be changed to *the front bench*, but the named entity recognition model fails to capture it and leads to missing it from the correction candidate.

Example 1) - Success

Evidence Sentences: Her husband, Philippe Braham, was one of 17 people killed in January's terror attacks in Paris. One month after the terror attacks in Paris, a gunman attacked a synagogue in Copenhagen, Denmark, killing Dan Uzan, who was working as a security guard for a bat mitzvah party.

Input Summary:
Valerie Braham was one of 17 people killed in January 's terror attacks in Paris

Corrected Summary: *Philippe Braham* was one of 17 people killed in January's terror attacks in Paris.

Top3 Correction Candidates for Valerie Braham:
Philippe Braham, Dan Uzan, bat mitzvah

Example 2) - Failure

Evidence Sentences: Sawyer Sweeten grew up before the eyes of millions as a child star on the endearing family sitcom " Everybody Loves Raymond." Sweeten , best known for his role Geoffrey Barone , was visiting family in Texas, entertainment industry magazine Hollywood Reporter reported, where he is believed to have shot himself on the front porch.

Input Summary:
He is believed to have shot himself on Raymond

Corrected Summary: He is believed to have shot himself on Raymond.

Top3 Correction Candidates for Raymond:
Everybody Loves Raymond, Geoffrey Barone, Sawyer Sweeten

Figure 3: Case study on our proposed factual error correction system. The entities in the evidence sentences are highlighted. The blue color on each entity in each input summary represents the *erroneous score*, and the darker the color, the higher the *erroneous score*.

5 Conclusion

In this paper, we proposed an efficient factual error correction system RFEC based on two retrieval steps. RFEC first retrieves evidence sentences based on textual similarities between the summary and the article for detecting and correcting factual errors. Then, if there is an entity that is a cause of factual errors, RFEC substitutes it with one of the entities in the evidence sentences as a retrieval-based approach. Experiments on two benchmark datasets demonstrate that our proposed method shows competitive results compared to strong baseline seq2seq with a much faster inference speed.

Acknowledgement

K. Jung is with Dept. of Electrical and Computer Engineering, Seoul National University, Seoul, Korea. This work was supported by AIRS Company in Hyundai Motor and Kia through HMC/KIA-SNU AI Consortium Fund.

References

Meng Cao, Yue Dong, Jiapeng Wu, and Jackie Chi Kit Cheung. 2020. Factual error correction for abstractive summarization models. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 6251–6258.

Ziqiang Cao, Furu Wei, Wenjie Li, and Sujian Li. 2018. Faithful to the original: Fact aware neural abstractive summarization. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 32.

Sihao Chen, Fan Zhang, Kazoo Sone, and Dan Roth. 2021. Improving faithfulness in abstractive summarization with contrast candidate generation and selection. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 5935–5941, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Tanya Goyal and Greg Durrett. 2021. Annotating and modeling fine-grained factuality in summarization. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 1449–1462.

Wojciech Kryscinski, Bryan McCann, Caiming Xiong, and Richard Socher. 2020. Evaluating the factual consistency of abstractive text summarization. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 9332–9346, Online. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 7871–7880.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In Text Summarization Branches Out, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Hui Lin and Vincent Ng. 2019. Abstractive summarization: A survey of the state of the art. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 33, pages 9815–9822.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 1906–1919.

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In Thirty-First AAAI Conference on Artificial Intelligence.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gulçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning, pages 280–290, Berlin, Germany. Association for Computational Linguistics.

Artidoro Pagnoni, Vidhisha Balachandran, and Yulia Tsvetkov. 2021. Understanding factuality in abstractive summarization with frank: A benchmark for factuality metrics. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 4812–4829.

James Thorne and Andreas Vlachos. 2021. Evidence-based factual error correction. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 3298–3309.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. Fever: a large-scale dataset for fact extraction and verification. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pages 809–819.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In International Conference on Learning Representations.

Zheng Zhao, Shay B. Cohen, and Bonnie Webber. 2020. Reducing quantity hallucinations in abstractive summarization. In Findings of the Association for Computational Linguistics: EMNLP 2020, pages 2237–2249, Online. Association for Computational Linguistics.

Chenguang Zhu, William Hinthorn, Ruochen Xu, Qingkai Zeng, Michael Zeng, Xuedong Huang, and Meng Jiang. 2021. Enhancing factual consistency of abstractive summarization. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 718–733.