BeLeaf: Belief Prediction as Tree Generation

John Murzaku[△◇], **Owen Rambow**[♠◇]
[△] Department of Computer Science [♠] Department of Linguistics
[◇] Institute for Advanced Computational Science
Stony Brook University, Stony Brook, NY, USA

Abstract

We present a novel approach to predicting source-and-target factuality by transforming it into a linearized tree generation task. Unlike previous work, our model and representation format fully account for the factuality tree structure, generating the full chain of nested sources instead of the last source only. Furthermore, our linearized tree representation significantly compresses the amount of tokens needed compared to other representations, allowing for fully end-to-end systems. We achieve state-of-the-art results on FactBank and the Modal Dependency Corpus, which are both corpora annotating source-and-target event factuality. Our results on fine-tuning validate the strong generality of the proposed linearized tree generation task, which can be easily adapted to other corpora with a similar structure. We then present BeLeaf, a system which directly leverages the linearized tree representation to create both sentence level and document level visualizations. Our system adds several missing pieces to the source-and-target factuality task such as coreference resolution and event head word to syntactic span conversion. Our demo code is available on https://github.com/yurpl/beleaf and our video is available on https://youtu.be/SpbMNnin-Po.

1 Introduction

The term "factuality" (or belief[1]) refers to what extent an event mentioned by the author or by sources in a text is presented as being factual. In other words, the task aims to predict whether the author or the mentioned sources in the text believes the event happened. The event factuality prediction task (EFP) has received a lot of attention over the past few years, but only in the perspective of the author of the text, disregarding the factuality of events according to all sources (Lee et al., 2015; Stanovsky et al., 2017; Rudinger et al., 2018; Pouran Ben Veyseh et al., 2019; Jiang and de Marneffe, 2021).

Two notable exceptions are the FactBank corpus (Saurí and Pustejovsky, 2009) and the Modal Dependency Parsing corpus (MDP) (Yao et al., 2021). Both corpora annotate event factuality according to the author of the text, and also according to the sources mentioned in the text, with some slight differences. FactBank represents factuality on the sentence level, while the MDP corpus represents factuality as a document-level modal dependency structures (MDS) proposed by Vigus et al. (2019). The MDP structure uses a tree representation where the author of the text (AUTHOR) is the root, and events and other sources are child nodes of the author. The corpora also differ slightly on labels: FactBank annotates the factuality of events (alongside their polarities) as CT (certain), PR (probable), PS (possible), UU (unknown), while the MDP corpus annotates events as full positive (Pos), partial positive (Prt), positive neutral (Neut), negative neutral (Neutneg), partial negative (Prtneg) and full negative (Neg).

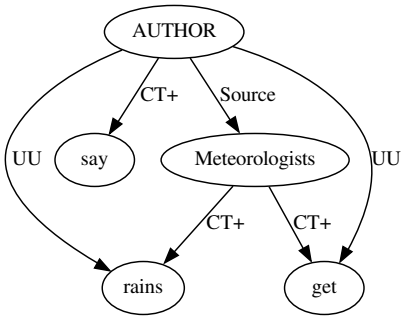

Figure 1: Source-and-target factuality represented as a modal dependency structure for the sentence "Meteorologists **say** the weather will **get** worse because there will be **rains**."

An NLP system's ability to accurately attribute events' factuality according to all sources is vital

[1]We use the terms interchangeably since our system is called BeLeaf. Factuality is closely related to the notion of "belief" as used in cognitive science and AI.

Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 3: System Demonstrations), pages 97–106

for downstream tasks that are based on those events. Consider the example sentence in Figure 1 where we have three events: the *say* event, the *rain* event, and the *get* event. We also have one source that the author mentions, *Meteorologists*. The author is certain (CT+) that the say event happened . However, the author does not tell us (UU) about their view of the factuality of the rains event or the get event because it is being presented by the source Meteorologists. According to the source Meteorologists, these events are factual (CT+). An information extraction system should extract the specific factuality of these events depending on all sources presenting the event, not only the author. Systems and humans can then make a separate judgement about the weather based on their sense of the trustworthiness of the author and of meteorologists.

In this paper, we present the source-and-target event factuality prediction task as a linearized tree generation task. We represent both FactBank and the MDP corpus as linearized trees, achieving state-of-the-art results for both corpora and beating both our FactBank results (Murzaku et al., 2023) and the MDP results from Yao et al. (2022). This representation format not only performs better, but also allows for a clear and interpretable visualization, which we show in our BeLeaf system.

2 Related Work

Author-Only Factuality All previous approaches to the event factuality prediction task were in the author-only setting, ignoring nested sources. Early approaches used rule-based systems and/or lexical and dependency tree based features (Nairn et al., 2006; Lotan et al., 2013). Early machine learning work used SVMs alongside dependency tree and lexical based features (Diab et al., 2009; Prabhakaran et al., 2010; Lee et al., 2015; Stanovsky et al., 2017). Neural work includes LSTMs with multi-task or single-task approaches (Rudinger et al., 2018) or using BERT representations alongside a graph convolutional neural network (Pouran Ben Veyseh et al., 2019). Jiang and de Marneffe (2021) expand on previous work by using other event factuality corpora in multiple training paradigms while also introducing a simpler architecture. These approaches evaluate on Pearson correlation and mean absolute error (MAE), failing to capture individual label performance and assuming events are given. We (Murzaku et al., 2022) provided the first end-to-end evaluation using F-measure and improve on FactBank.

Source and Target Factuality One of the main corpora experimented on in this paper, which annotates all events introduced in a corpus of exclusively newswire text is the FactBank corpus (Saurí and Pustejovsky, 2009).The FactBank corpus not only annotates the factuality presented by the author of a text towards an event, but also the factuality of events according to their presentation by sources mentioned inside of the text. Saurí and Pustejovsky (2012) were the first to investigate and perform experiments on the source and target annotations in FactBank. Their evaluation was not end-to-end and was given manual annotations, so it is therefore not comparable to our results on FactBank. We (Murzaku et al., 2023) were the first to represent the event factuality prediction task as a generation task using Flan-T5 while also accounting for source and target factuality. However, our previous model did not account for the full nesting structure of the source since our model only generated the last nested source.

Our new system generates the full nesting structure, and is therefore not comparable to our previous FactBank results as that task was far easier and incomplete. Yao et al. (2021) also propose a source-and-target corpus (MDP corpus) and Yao et al. (2022) improve on their previous results by using a prompt-based approach where they treat factuality prediction as a BIO tagging task, fine-tuning on XLM-RoBERTa (Conneau et al., 2020). Following the modal dependency structure from Vigus et al. (2019), their corpus annotates events, sources, and credibility of sources throughout a whole document. The top level source is always the author of the text. While similar to FactBank in some ways, there are some key differences (which we describe in Section 3.1), making the corpora incompatible for joint experiments with FactBank.

Document Level Factuality Qian et al. (2019) are the first to present the document level factuality task, but again in the author-only setting. Their work is expanded by Cao et al. (2021), Qian et al. (2022), and more recently Zhang et al. (2023). In this task the input is a document and a factuality target, and the output is the label representing the factuality attributed by the author to the provided target. Our task is different, which is to find all sources and targets of factuality assessments.

Our work differs from the previous work on

event factuality prediction in two major ways:
(i) We are the first to provide a novel and state-of-the-art machine learning representation for the source-and-target event factuality prediction tasks (both sentence level and document level).
(ii) To our knowledge, we are the first to provide a unified toolkit and intuitive front-end interface for the event factuality prediction task. Our toolkit improves on several shortcomings of previous corpora and approaches to this task and our interface leverages the new tree representation for a clear and interpretable visualization.

3 Approach

3.1 Data Representation

FactBank The FactBank corpus annotates event factuality according to the author and sources attributied by the author. When a source is not present or explicitly mentioned in text, FactBank uses the *GEN* label. For example, in the sentence *The transaction is expected to close*, there is no explicit mention of a source attributing the events, therefore being labeled *GEN*. When a sentence contains a fragment of a quotation, FactBank uses the DUMMY label. We represent all sources including *GEN* and *DUMMY*.

MDP corpus The MDP corpus also annotates factuality of events according to the author and nested sources. Additionally, the MDP corpus annotates the factuality between the author and embedded sources (or further embedded sources) to account for overall credibility of sources by attributing the author's certainty towards them. For example, in Figure 1, the MDP corpus would annotate the edge between AUTHOR and Meteorologists as *Pos*, or full positive, meaning the author is certain the Meteorologists are presenting an event. In our linearized tree representation, we include these factuality labels when beginning a new nest to also capture credibility. Finally, like *GEN* in FactBank, the MDP corpus uses NULL to capture sources that are not present or explicitly mentioned in text.

Tree Generation We approach the source-and-target event factuality prediction task as a linearized tree generation task. Consider the example sentence from Figure 1 in a FactBank format. We reformat the FactBank data as the following input/output pair for machine learning:
Input: Meteorologists say the weather will get worse because more rains are on the way.

Pos	Prtpos	Prtneg	Neg
true	ptrue	pfalse	false

Table 1: Factuality values for the MDP corpus

CT+	PR+	UU	PR-	CT-
true	ptrue	unknown	pfalse	false

Table 2: Factuality values for the FactBank corpus

Output Tree: (AUTHOR (rains unknown) (get unknown) (say true) (Meteorologists nest (rains true) (get true)))

We add the special *nest* token to denote the beginning of a nested source and their respective presentation of events.

3.2 Labels

In Section 1, we present the corpus-specific labels. The labels are as follows:
Certain: Corresponding to FactBank CT±, MDP Pos/Neg. Here, the author commits to the truth or falseness of the presented situation.
Probable: Corresponding to FactBank PR±, MDP Prtpos/Prtneg. Here, the author presents the situation as probable.
Fully underspecified Corresponding to FactBank UU. The source does not know what is the factual status of the event, or does not commit a belief it.

For our linearized tree generations, we convert each label to distinct words. For FactBank, we follow our previous FactBank work (Murzaku et al., 2022) and collapse the PR+/PS+ and PR-/PS- labels. Similarly, for the MDP corpus we follow Yao et al. (2022) and collapse the Prt/Neut and Prtneg/Neutneg labels. Table 1 and Table 2 show our mapped values for the MDP and FactBank corpora respectively.

3.3 Model

We use the encoder-decoder pre-trained Flan-T5 model (Chung et al., 2022) and the decoder only GPT-3 model (Brown et al., 2020). The Flan-T5 model is an instruction fine-tuned model with significant performance improvements compared to T5 (Raffel et al., 2020) and better adaptability to unseen tasks as a result of instruction tuning. Furthermore, the larger parameter variants of Flan-T5 have comparable or better performance on some tasks to GPT-3. By formulating the linearized tree construction as a generation task, our models are

	MiF1	AMiF1	AMF1	CT+	PR+	UU	PR-	CT-
Murzaku et al. (2022)	-	-	0.680	0.767	**0.714**	0.735	**0.667**	0.519
Murzaku et al. (2023)*	0.645	0.740	0.616	**0.815**	0.456	0.717	0.444	0.646
Flan-T5-Tree (Ours)	**0.695**	**0.766**	**0.708**	0.805	0.587	**0.752**	**0.667**	**0.733**
GPT-3-Tree (Ours)	0.658	0.760	0.678	0.778	0.455	0.747	**0.667**	0.723

Table 3: Results on the FactBank corpus for our Flan-T5 and GPT-3 systems evaluating on micro-f1 (MiF1), author micro-f1 (AMiF1), author macro-f1 (AMF1), and author per-label f1. We show baseline results from Murzaku et al. (2022) and redo Murzaku et al. (2023) for direct comparison (signaled by *). A shaded cell indicates state-of-the-art and statistically significant ($p < 0.05$)

	dev	test
Yao et al. (2021) P	0.697	0.675
Yao et al. (2021) J	0.703	0.690
Yao et al. (2022)	0.727	0.719
Flan-T5-Tree (Ours)	0.762	**0.749**
GPT-3-Tree (Ours)	**0.764**	0.741

Table 4: Results on the MDP corpus evaluated on micro-f1 compared to previous state-of-the-art results from Yao et al. (2022)

end-to-end and do not need gold event words as input.

4 Experiments: Fine-tuning

4.1 Corpora

We use our split of FactBank (Murzaku et al., 2022) for all examples including author and non-author sources. We also use the MDP corpus split from Yao et al. (2021). Like Yao et al. (2021) and Yao et al. (2022), we only consider examples with two levels of sources. For FactBank, we consider all levels of sources, but the majority have between one and three levels, with only four examples having three levels of sources.

4.2 Experiment Details

We use a standard fine-tuning approach on Flan-T5 and GPT-3. We fine-tune our Flan-T5 models for at most 20 epochs with a learning rate of 3e-4, with early stopping being used if the validation micro-F1 did not increase. We use task-specific prefixes and note that using instructions did not boost performance. Our Flan-T5 experiments are averaged over three runs using fixed seeds. We perform significance testing to previous baselines using a paired t-test. Due to costs, our GPT-3 experiments are performed once. We leave more experimental details to Appendix B.

4.3 Evaluation

We evaluate on micro-f1 (MiF), author-only micro-f1 (AMiF1), and author-only macro-f1 (AMF1) for FactBank. All of these metrics help us quantify to what extent we capture the full author and non-author sources in our generations: MiF1 shows how well we can generate full tree structures including their nesting, AMiF1 shows how well our model characterizes events only from the perspective of the author (which is a majority of events), and AMF1 shows how well we predict *all* factuality labels regardless of frequency, according to the author. For the modal dependency corpus, we follow Yao et al. (2022) evaluating on micro-f1.

4.4 Results: Fine-tuning

FactBank Table 3 shows results for our linearized tree generation model on the FactBank corpus. We compare our results to our baselines from Murzaku et al. (2022) and Murzaku et al. (2023). Murzaku et al. (2023) do not generate nested sources. We modify our baseline to generate all sources by adding the full nestings to their source-and-target triplet generation task. For example, a doubly nested triplet (Mary, said, true) becomes (AUTHOR_John_Mary, said, true). Our Flan-T5 system outperforms the previous state-of-the-art results and GPT-3 on all micro-f1, author-only micro-f1, author macro-f1. Furthermore, on the per-label f-measures, we see the largest boost and new SOTA in the CT- label (9% absolute increase), and slight but statistically significant increase in the UU label.

MDP Corpus Table 4 shows results for our linearized tree generation models on the MDP corpus. We beat the previous state of the art from Yao et al. (2022) on dev by 3.7% and on test by 3%. We observe that on test, fine-tuning Flan-T5 outperforms fine-tuning GPT-3, which can be explained

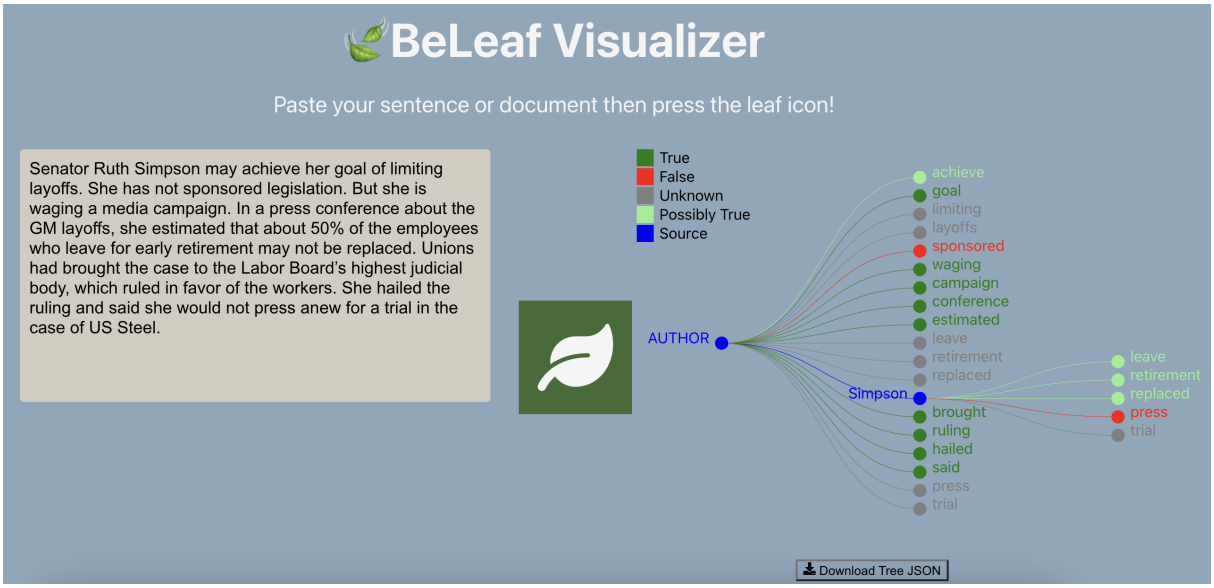

Figure 2: The BeLeaf system with a textbox for sentence or document inputs, the leaf button to begin inference, and our output tree with corresponding belief values as edge colors and labels.

by Flan-T5's generalizability to unseen tasks from instruction tuning.

5 BeLeaf: System Description

In this section, we present our BeLeaf system which leverages our generated tree structure. Our system is split into three parts: a generalized API for querying our Flan-T5 model with either sentences or documents, a preprocessing pipeline where we improve on the document level event factuality/belief task from Yao et al. (2021) by accounting for coreference, and a postprocessing pipeline accounting for syntactic spans with a tree visualization tool. Our system is shown in Figure 2.

5.1 API

We build a REST API using Flask (Grinberg, 2018), adding a single inference endpoint for all inference. Our API then queries our top-performing Flan-T5 model fine-tuned on FactBank. Before beginning inference, we perform a preprocessing pipeline.

5.2 Preprocessing

To account for both sentence and document level belief, we use spaCy (Honnibal and Montani, 2017) for splitting our model into sentences, and then pass this into our sentence-level FactBank model. This allows us to maximize our systems speed but we still need to account for beliefs across sentences. Therefore, to create a true document level belief system, we add a coreference resolver in our system. The MDP Corpus (Yao et al., 2021) is not a true document level representation of belief since they do not account for coreference resolution, and therefore a source can be repeated. We use the fastcoref library (Otmazgin et al., 2022) to perform coreference which was found to maximize speed with a minimal drop in accuracy for the coreference resolution task.

5.3 Postprocessing and Tree Visualization

Postprocessing After we get an output from our model, we perform a postprocessing pipeline to get syntactic spans. Since both FactBank and the MDP corpora annotate only syntactic head words or noun events, we oftentimes miss the full syntactic span and context of the event in question. To address this, we create a head-to-span module that uses spaCy (Honnibal and Montani, 2017) to return the full syntactic span. We include this representation as a hover-over in the tree visualization and also include it as a data download option.

Tree Visualization The final piece of our system is our tree visualization module. A sample output of our tree output is shown in the right hand side of Figure 2. To clearly distinguish between nested sources and their child events, we do not visualize with a DAG structure like the representation in Figure 1 where edges connect to nodes from both the author and the nested source, but rather a distinguished-source tree structure. All visualizations are made in JavaScript using the d3 library (Bostock, 2012). Furthermore, to allow researchers

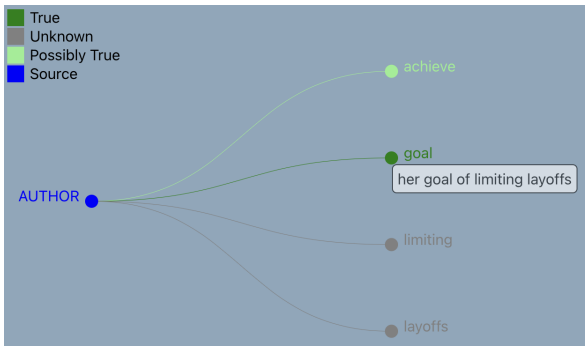

Figure 3: Sentence level output including syntactic span labels.

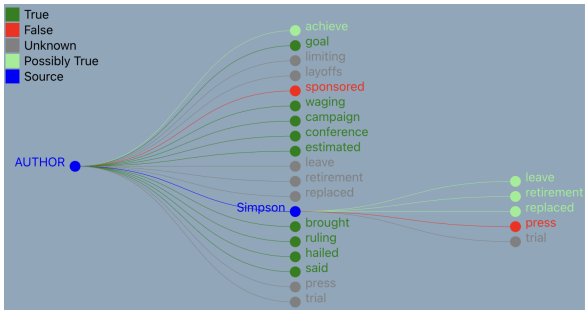

Figure 4: Document level output with a nested source Simpson.

and users of our package to utilize the tree information, we include a download data button that returns the full tree in JSON format. This JSON file includes all nodes with their parent/child structure and events as syntactic spans or heads with their corresponding belief values.

6 BeLeaf: Output and Visualization

In this section, we provide examples for both sentence level and document level belief, alongside their corresponding tree outputs and JSON representations.

Sentence Level Consider the following sentence:

```
Senator Ruth Simpson may achieve her
goal of limiting layoffs.
```

Here, the author presents multiple events: achieve, goal, limiting, layoffs. Note that in FactBank, an event can also be a noun, which is why goal and layoffs are included. Figure 3 shows the tree structure and a hover-over syntactic span from our head-to-span output.

Document Level We now expand the previous example to show a short document level output, including coreference and nested sources:

```
Senator Ruth Simpson may achieve
her goal of limiting layoffs.  She
has not sponsored legislation.  But
she is waging a media campaign.  In
a press conference about the GM
layoffs, she estimated that about 50%
of the employees who leave for early
retirement may not be replaced.  Unions
had brought the case to the Labor
Board's highest judicial body, which
ruled in favor of the workers.  She
hailed the ruling and said she would
not press anew for a trial in the case
of US Steel.
```

Our output is shown in Figure 4. Our system correctly coreferences the pronoun she with Senator Ruth Simpson, tracking her presentation of events throughout this document. Furthermore, this example effectively visualizes the nested belief/source-and-targetr factuality structure. For example, we see the perspective of events leave, retirement, replaced, press, and trial according to both the author and according to Senator Ruth Simpson.

Output JSON As shown in Figure 2, our system also includeds a button to download a JSON formatted tree structure. Using our document level example, we show a shortened example output:

```
{
  "name": "AUTHOR",
  "children": [
    {
      "name": "retirement",
      "belief": "unknown",
      "synSpan": "early retirement",
      "children": []
    },
    ...,
    {
      "name": "Simpson",
      "children": [
        {
          "name": "retirement",
          "belief": "possibly true",
          "synSpan": "early retirement",
          "children": []
        },
        ...,
      ]
    }
  ]
}
```

7 Conclusion

We propose a linearized tree generation model for the source-and-target event factuality task prediction using Flan-T5 and GPT-3. We evaluate the model on FactBank and the MDP corpus, and achieve results for both. With our new representation and state of the art Flan-T5 system, we

present BeLeaf, a system for both sentence and document level factuality. We provide a preprocessing pipeline that accounts for coreference to create true document level representations of factuality. An inference API is then made which feeds to a postprocessing pipeline that creates syntactic spans from head words for users to see the full event contexts. Finally, we merge everything into a tree visualization software that also includes a data download option.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Award No. 2125295 (NRT-HDR: Detecting and Addressing Bias in Data, Humans, and Institutions). Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

We thank the Institute for AI-Driven Discovery and Innovation at Stony Brook University for access to the computing resources needed for this work. These resources were made possible by NSF grant No. 1919752 (Major Research Infrastructure program).

Limitations

We note that all experiments are performed on only two English source-and-target event factuality corpora. While we achieve state-of-the-art results for English, we do not know how well our linearized tree generation model can generalize to other languages. We will investigate multilingual source-and-target event factuality prediction as linearized tree generation in future work.

For our GPT-3 experiments, we only perform one run and therefore do not report an average over 3 runs. We do this to minimize costs.

We note that these experiments do not account for potential biases prevalent in fine-tuning large language models. We hypothesize that for some sources in text (i.e. power figures, authorities, or specific names), there may be biases towards certain factuality labels. We will investigate these biases in future work because an event factuality prediction system with inherent bias can have real world consequences.

Ethics Statement

Our paper is foundational research and we are not tied to any direct applications. However, our experiments do not account for potential biases prevalent in fine-tuning large language models. In a real world deployment of our model, we hypothesize that there could be a potential mislabelling of factuality values depending on bias towards sources of utterances. For example, if a power figure states an event, will the event label be more biased towards being factual just because of the source of the statement? Furthermore, are large language models biased in predicting or failing to predict specific nested sources? For example, are certain groups, names, or specific sources being ignored? Finally, how much of a role does our new representation format contribute to bias? We will investigate these questions and issues in future work.

References

Mike Bostock. 2012. D3.js - data-driven documents.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Pengfei Cao, Yubo Chen, Yuqing Yang, Kang Liu, and Jun Zhao. 2021. Uncertain local-to-global networks for document-level event factuality identification. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2636–2645, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Mona Diab, Lori Levin, Teruko Mitamura, Owen Rambow, Vinodkumar Prabhakaran, and Weiwei Guo. 2009. Committed belief annotation and tagging. In *Proceedings of the Third Linguistic Annotation Workshop (LAW III)*, pages 68–73, Suntec, Singapore. Association for Computational Linguistics.

William Falcon et al. 2019. Pytorch lightning. *GitHub. Note: https://github. com/PyTorchLightning/pytorch-lightning*, 3(6).

Miguel Grinberg. 2018. *Flask web development: developing web applications with python*. " O'Reilly Media, Inc.".

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Nanjiang Jiang and Marie-Catherine de Marneffe. 2021. He thinks he knows better than the doctors: BERT for event factuality fails on pragmatics. *Transactions of the Association for Computational Linguistics*, 9:1081–1097.

Kenton Lee, Yoav Artzi, Yejin Choi, and Luke Zettlemoyer. 2015. Event detection and factuality assessment with non-expert supervision. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1643–1648, Lisbon, Portugal. Association for Computational Linguistics.

Amnon Lotan, Asher Stern, and Ido Dagan. 2013. TruthTeller: Annotating predicate truth. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 752–757, Atlanta, Georgia. Association for Computational Linguistics.

John Murzaku, Tyler Osborne, Amittai Aviram, and Owen Rambow. 2023. Towards generative event factuality prediction. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 701–715, Toronto, Canada. Association for Computational Linguistics.

John Murzaku, Peter Zeng, Magdalena Markowska, and Owen Rambow. 2022. Re-examining FactBank: Predicting the author's presentation of factuality. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 786–796, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Rowan Nairn, Cleo Condoravdi, and Lauri Karttunen. 2006. Computing relative polarity for textual inference. In *Proceedings of the Fifth International Workshop on Inference in Computational Semantics (ICoS-5)*.

Shon Otmazgin, Arie Cattan, and Yoav Goldberg. 2022. F-coref: Fast, accurate and easy to use coreference resolution. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 48–56, Taipei, Taiwan. Association for Computational Linguistics.

Amir Pouran Ben Veyseh, Thien Huu Nguyen, and Dejing Dou. 2019. Graph based neural networks for event factuality prediction using syntactic and semantic structures. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4393–4399, Florence, Italy. Association for Computational Linguistics.

Vinodkumar Prabhakaran, Owen Rambow, and Mona Diab. 2010. Automatic committed belief tagging. In *Coling 2010: Posters*, pages 1014–1022, Beijing, China. Coling 2010 Organizing Committee.

Zhong Qian, Peifeng Li, Qiaoming Zhu, and Guodong Zhou. 2019. Document-level event factuality identification via adversarial neural network. 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 2799–2809, Minneapolis, Minnesota. Association for Computational Linguistics.

Zhong Qian, Heng Zhang, Peifeng Li, Qiaoming Zhu, and Guodong Zhou. 2022. Document-level event factuality identification via machine reading comprehension frameworks with transfer learning. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 2622–2632, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Rachel Rudinger, Aaron Steven White, and Benjamin Van Durme. 2018. Neural models of factuality. 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 731–744, New Orleans, Louisiana. Association for Computational Linguistics.

Roser Saurí and James Pustejovsky. 2009. Factbank: a corpus annotated with event factuality. *Language resources and evaluation*, 43:227–268.

Roser Saurí and James Pustejovsky. 2012. Are you sure that this happened? assessing the factuality degree of events in text. *Computational Linguistics*, 38(2):261–299.

Gabriel Stanovsky, Judith Eckle-Kohler, Yevgeniy Puzikov, Ido Dagan, and Iryna Gurevych. 2017. Integrating deep linguistic features in factuality prediction over unified datasets. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 352–357, Vancouver, Canada. Association for Computational Linguistics.

Meagan Vigus, Jens E. L. Van Gysel, and William Croft. 2019. A dependency structure annotation for modality. In *Proceedings of the First International Workshop on Designing Meaning Representations*, pages

182–198, Florence, Italy. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Jiarui Yao, Haoling Qiu, Jin Zhao, Bonan Min, and Nianwen Xue. 2021. Factuality assessment as modal dependency parsing. 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 1540–1550, Online. Association for Computational Linguistics.

Jiarui Yao, Nianwen Xue, and Bonan Min. 2022. Modal dependency parsing via language model priming. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2913–2919, Seattle, United States. Association for Computational Linguistics.

Heng Zhang, Peifeng Li, Zhong Qian, and Xiaoxu Zhu. 2023. Incorporating factuality inference to identify document-level event factuality. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13990–14002, Toronto, Canada. Association for Computational Linguistics.

A Data

FactBank We split our corpus using the same split and methods as Murzaku et al. (2022), which also includes splitting by article. We follow a similar evaluation setup evaluating on macro-f1 and per-label f1. The FactBank corpus can be obtained by researchers from the Linguistic Data Consortium, catalog number LDC2009T23.

Modal Dependency Corpus We use the modal dependency corpus from Yao et al. (2022). We follow the same evaluation setup and procedure evaluating on micro-f1.

Tree generation We reformat the FactBank data for our machine learning representation. All linearized trees have the author of the text as the root. We add the special token **nest** to declare nestings according to a source. We show the following example and its linearized tree:

Input: Meteorologists say the weather will get worse because more rains are on the way.

Tree: (Author (rains unknown) (get unknown) (say true) (Meteorologists **nest** (rains true) (get true)))

	train	dev	test
FactBank	8,153	2,345	1,165
MDP	21,855	2,605	2,464

Table 5: Number of examples (sum of sources and events) in the splits for each corpus.

B Details on Experiments

All experiments besides our GPT-3 experiments used our employer's GPU cluster. We performed experiments on a Tesla V100-SXM2 GPU. Compute jobs typically ranged from 30 minutes for standard fine-tuning experiments to 50 minutes for synthetic data generation. We do not do any hyperparameter search or hyperparameter tuning.

FactBank experiments We fine-tuned our models for at most 10 epochs, with early stopping being used if the macro-F1 did not increase for 3. We use a standard fine-tuning approach with Flan-T5-large which has 780 million parameters. We also experimented with Flan-T5-xl which has 3 billion parameters, but often ran into memory issues due to heavy GPU load. We use the Adafactor optimizer along with a Adafactor scheduler, which dynamically adapts the learning rate throughout the training process to ensure optimal model performance. All metrics for experiments were averaged over three runs using fixed seeds (7, 21, and 42). We report the average over three runs and the standard deviation over three runs.

Modal dependency corpus experiments We fine-tuned our models for at most 20 epochs, with early stopping being used if the micro-F1 did not increase for 20 epochs. We use a standard fine-tuning approach with Flan-T5-large which has 780 million parameters. We use the Adafactor optimizer along with a Adafactor scheduler, which dynamically adapts the learning rate throughout the training process to ensure optimal model performance. All metrics for experiments were averaged over three runs using fixed seeds (7, 21, and 42). We report the average over three runs and the standard deviation over three runs.

GPT-3 experiments We used a standard fine-tuning approach using the OpenAI API. We used a temperature of 0.0 for all experiments to select the most likely token at each step. Because of fine-tuning costs, we perform only one run and therefore do not report standard deviation.

Packages To fine-tune our models and run experiments, we used PyTorch lightning Falcon et al. (2019) and the transformers library provided by HuggingFace Wolf et al. (2019). All code for fine-tuning, modelling, and pre-processing will be made available.

Corpus Splits Table 5 shows the train-dev-test splits for FactBank and the MDP corpus respectively.