Factuality Assessment as Modal Dependency Parsing

Jiarui Yao[1], Haoling Qiu[2], Jin Zhao[1], Bonan Min[2], and Nianwen Xue[1]

[1]Brandeis University
{jryao, jinzhao, xuen}@brandeis.edu
[2]Raytheon BBN Technologies
{haoling.qiu, bonan.min}@raytheon.com

Abstract

As the sources of information that we consume everyday rapidly diversify, it is becoming increasingly important to develop NLP tools that help to evaluate the credibility of the information we receive. A critical step towards this goal is to determine the factuality of events in text. In this paper, we frame factuality assessment as a *modal dependency* parsing task that identifies the events and their sources, formally known as *conceivers*, and then determine the level of certainty that the sources are asserting with respect to the events. We crowdsource the first large-scale data set annotated with modal dependency structures that consists of 353 Covid-19 related news articles, 24,016 events, and 2,938 conceivers.[1] We also develop the first modal dependency parser that jointly extracts events, conceivers and constructs the modal dependency structure of a text. We evaluate the joint model against a pipeline model and demonstrate the advantage of the joint model in conceiver extraction and modal dependency structure construction when events and conceivers are automatically extracted. We believe the dataset and the models will be a valuable resource for a whole host of NLP applications such as fact checking and rumor detection.

1 Introduction

"*We're not just fighting an epidemic; we're fighting an infodemic.*"

— Tedros Adhanom, WHO

The ongoing COVID-19 pandemic taught us the importance of determining factuality of events at a time when the sources of media we consume have greatly diversified. This is compounded by the fact that the information that we receive from these news sources usually does not go through as a rigorous editing and verification process as traditional media do. The sheer volume of the new media content makes human verification impossible and there is thus an increasing need for NLP tools that help verify statements made in these media sources.

To verify if an event has indeed happened we first need to determine the level of certainty with which the event is asserted by the information source, which defaults to the author of a document but can also be another source in the text that the author attributes the information to. The factuality of an event cannot be fully determined without also taking into account the credibility of information source. Consider the text snippet in (1):

(1) WBUR: A man in his 20s from Worcester County **tested positive** Tuesday for the new, apparently more contagious coronavirus variant, public health officials **said**. The variant was first **detected** in the United Kingdom, and experts have **warned** that it could soon **become widespread** in the U.S.

Suppose our goal here is to determine the factuality of the statements in (1). We first need to determine the level of certainty with which the source is committed to the factuality of the statements. While the "public health officials" are fairly certain that a man from Worcester County tested positive for the coronavirus variant, the "experts" were not as certain that the virus variant will definitely become widespread, as indicated by the linguistic cues like "could". In other words, there are different levels of certainty with which the two events are asserted. In addition, the credibility of information source is also crucially important when evaluating the factuality of the events (De Marneffe et al., 2012). If the information source is not "public health officials" and instead it is an anonymous source, the information that the Worcester

[1]https://github.com/Jryao/modal_dependency

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, pages 1540–1550
August 1–6, 2021.

man tested positive will be less credible. In fact, the factuality of the events also depends on the WBUR, the "author" of this text. If the author made up these statements, then the Worcester man testing positive would not be a fact, regardless of the level of certainty with which the events are asserted. Ultimately, it is impossible to fully determine the credibility of a source purely based on the information within a single text, but linking each event to its source or chain of sources allows us to verify the factuality of the event against other sources and our world knowledge. Therefore, identifying the level of certainty with which is an event is asserted together with its source is a crucial first step in assessing the factuality of the event.

Previous work on factuality assessment has focused on determining the level of certainty that is asserted on events and framed it as a classification or regression problem (Saurí and Pustejovsky, 2012; Lee et al., 2015; Stanovsky et al., 2017; Rudinger et al., 2018; Qian et al., 2018). However, as we discussed above, the level of certainty alone is insufficient in determining the factuality of an event. In this work, we adopt a factuality representation framework proposed in (Vigus et al., 2019) called *modal dependency structure* (MDS). A modal dependency structure is formally a document-level structure where nodes are events and sources, known as *conceivers* while edges represent the *modal strength*, or the level of certainty that the conceiver holds towards an event. Figure 1 shows the modal dependency structure of the text in (1).

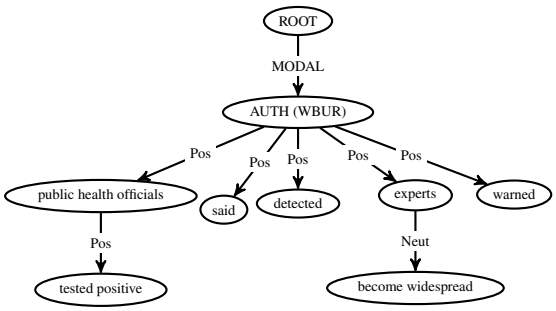

Figure 1: A modal dependency tree for example 1.

One main advantage of MDS over previous approaches to factuality is that an MDS explicitly represents both the conceiver and the event and represents the modal strength as the level of certainty that the conceiver holds towards the events. It is also a hierarchical structure that allows nested representations when multiple sources are possible. For example, in Figure 1, the factuality of the *tested positive* event depends on both the credibility of "public health officials", as well as the author (AUTH) of this text.

Representing factuality as a modal dependency structure also allows us to cast factuality assessment as a modal dependency parsing problem, and opens up the door for using various structured prediction approaches to tackle this problem. Since no large-scale data annotated with MDS exists, we first need to develop a data set annotated with modal dependency structures to train and evaluate modal dependency parsing models. The main contributions of this work are as follows:

- We construct a large corpus annotated with modal dependency structures via crowdsourcing. It consists of 353 Covid-19 related news articles, in which 24,016 events and 2,938 conceivers are annotated. To our best knowledge, this corpus is the first large-scale corpus annotated with modal dependency structures.
- Although modal dependency structure is a complicated representation, we show that our data set is annotated with high consistency. We believe the crowdsourcing techniques we have developed will add to the knowledge of how to develop large-scale data sets via crowdsourcing, especially for complicated representations.
- We develop a joint modal dependency parsing model that extracts events, conceivers and parses a document into its modal dependency structure. We present experimental results that show the effectiveness of our parsing approach and the consistency of the data set. In addition, we evaluate the joint model against the pipeline model, and show the advantage of the joint model in overall end-to-end modal dependency parsing performance.

2 Acquiring a Modal Dependency Data Set

In this section we first provide additional detail for the modal dependency structure, and then present our strategy of decomposing the modal dependency structure into subtasks that are suitable for crowdsourcing. We also evaluate the quality of our annotated data set, and provide statistics relevant for training MDS parsing models.

2.1 Modal Dependency Structure

The modal dependency structure builds on the annotation scheme of FactBank (Saurí and Pustejovsky, 2009) and is inspired by the structured approach in temporal dependency annotation in Zhang and Xue (2018b). Like FactBank, the modal dependency structure combines epistemic strength *full, partial, neutral* with polarity values *positive, negative* to define a set of six values for modal strength. Table 1 shows the modal strength values (i.e. labels) used in modal dependency structures and their corresponding values in FactBank. As illustrated in Figure 1, these values are represented as edge labels in the modal dependency structure. Readers are referred to (Vigus et al., 2019) for how these six values are defined.

Modal Dependency	FactBank
full positive (Pos)	CT+
partial positive (Prt)	PR+
neutral positive (Neut)	PS+
neutral negative (Neutneg)	PS-
partial negative (Prtneg)	PR-
full negative (Neg)	CT-

Table 1: Modal strength values in the modal dependency structure and FactBank.

While Vigus et al. (2019) show that modal dependency structures (MDS) can be annotated with high inter-annotator agreement by expert annotators in a pilot annotation of six documents, a corpus that is much larger in scale is needed in order to train modal dependency parsers that can be used for downstream applications.

2.2 Crowdsourcing Modal Dependency Structures

To make MDS feasible for crowdsourcing, we have adopted a number of strategies. The first strategy is to decompose MDS annotation into four subtasks: event identification, event attachment, event modal strength annotation and modal superstructure construction. The instructions to crowd-workers for each subtask are piloted to ensure that the subtask can be performed with high consistency before they are set up for productive annotation. Second, where possible, we have applied a number of pre-processing steps to simplify the tasks for crowd-workers. In addition, we have also adopted a payment structure to incentivize high-quality work. In all subtasks, we require three crowd-workers to complete one assignment and use the majority vote answer as the final decision. All annotations are conducted on the Amazon Mechanic Turk platform.

Task 1: Event Identification Event annotation involves identifying event trigger words, which are typically verbs and nouns. We first extract event candidates using a publicly available, common event trigger word dictionary.[2] We then ran the Stanford CoreNLP dependency parser (Manning et al., 2014) on raw text to extract the verbs and the root of each syntactic dependency parse as candidate events. A pilot study shows that 90% of the verbs in the extracted candidate events are event triggers, so we decide to treat all verbs as event triggers and launch an event identification task for about 10K non-verb event candidates. We present crowd-workers with event candidates and ask them to decide if they are events.

Task 2: Event Attachment The next subtask is to attach a child event to a *parent*, which can be a conceiver (2a), another event (2b), or in the case of hypothetical situations, an abstract *have-condition* node (2c). To simplify things for crowd workers, we made the decision not to introduce abstract nodes in the modal dependency tree. Events in hypothetical situation are annotated as neutral events and attached to their conceivers directly.

A child event is attached to a parent event only when the parent event is a modal predicate. For example, in (2b), the parent of *visit* is *wants*. The modal predicates form a closed set and can be extracted with a list of modal event triggers. Using a dependency parser, we can reliably identify events that should be attached to modal events, so we can do this part of the annotation without soliciting judgments from crowd workers.

(2) a. A 72-year-old man **died**, the *police* said.
Pos (died, police)

b. John **wants** to **visit** Japan.
Neut (visit, wants)

c. If it **rains** tomorrow, I will stay at home.
Neut (have-condition, Author)
Pos (rains, have-condition)

In the majority of cases, the parent of an event is a conceiver (or the Author), as in (2a), where the conceiver of *died* is the *police*. For any given

[2] https://github.com/Jryao/temporal_dependency_graphs_crowdsourcing

event, the list of candidate conceivers can be very large, so some filtering is needed to shrink it down so that a smaller list of candidates is presented to crowd-workers.

To collect possible conceivers, we first construct a list of common conceiver-introducing predicates (CIPs) following Saurí and Pustejovsky (2009) and Vigus et al. (2019). Then, we extract possible conceivers with the Stanford CoreNLP parser from three sources: the subject of common CIPs in our list, the subject of all other events, and named entities that are possible conceivers, such as organization, person. For each event, we limited the candidate conceivers to those in the same paragraph as the event, and further filter out unlikely conceivers by their hypernyms in Wordnet.[3] We present a list of possible conceivers and ask workers to select the most appropriate one for the event in question.

Task 3: Event Modal Strength Annotation After attaching the events to their parent, the third task is to annotate modal strength from the conceiver's perspective, which are edge labels in modal dependency structures. Vigus et al. (2019) define six modal strength values listed in Table 1. In our pilot annotation, however, we found partial negative and neutral negative events only account for less than 2% of all events. To have a manageable crowdsourcing task and given their low frequency, we decide to merge partial negative and neutral negative events to negative events, and only use four labels: full positive, partial positive, neutral positive and negative.

The event modal strength task is annotated in two steps. In the first step, events are classified into three classes: full positive, negative, and neither. In the second step, events in the third class are further classified into partial positive and neutral positive. For example, (3a) is annotated as a full positive event, (3c) is annotated as a negative event, and (3b) is annotated as neither in the first step. (3b) is further labeled as a neutral positive event in the second step.

(3) a. The dog **barked**.
b. The dog might have **barked**.
c. The dog didn't **bark**.

Task 4: Conceiver Superstructure Construction The parent of a conceiver is the Author in the majority of cases, but it could also be another conceiver in some cases. (4a) and (4b) are two common cases where the parent of a conceiver is another conceiver. In (4a), the conceiver *Mary* and the embedded conceiver *John* are in the same sentence, and in (4b), the conceiver *John* is in quoted speech. For the cases like (4b), the two conceivers are not necessarily in the same sentence, but they are usually close to each other in the text.

(4) a. **Mary** says **John** wants to visit Japan.
Pos (John, Mary)
b. "**John** wants to visit Japan. He wants to go next summer." **Mary** says.
Pos (John, Mary)

Conceivers that don't have any neighboring conceivers are directly attached to the Author.[4] For the rest, we design a conceiver attachment task similar to the event attachment task. The modal strength of conceivers is decided by the modal strength of their CIPs, which is available after Task 3. For the conceivers that don't have an associated CIP, such as named entities, we ask crowd-workers to annotate their modal strength.

2.3 Quality Control Strategy

Our basic quality control strategy involves using two tests to select crowd-workers: a qualification test and a survival test. Workers need to first pass the qualification test in order to be eligible for working on a task. In addition, test questions with ground truth answers are embedded in each HIT. Workers need to maintain a high cumulative accuracy through the annotation process to remain eligible for the task. For the event identification subtask, our qualification test threshold and survival test threshold are both set to 80% accuracy. For the event attachment task, they are both set to 70% as it is a more challenging task.

We also developed a stratified payment approach to incentivize high-quality work. There is no guarantee that workers who have passed the qualification test will continue to perform well in the actual annotation task. To incentivize high-quality annotation, we adopt a stratified payment approach for event modal strength annotation. We offer a base payment of $ 0.01 per question, and increase it to $ 0.02 if the worker achieves a 70% accuracy on the test questions in that HIT, and further increase it to $ 0.03 if the worker achieves a 90% accuracy. The

[3] https://wordnet.princeton.edu

[4] In practice, if there is no other conceivers in the same paragraph, we attach that conceiver to the Author.

additional payment is paid using the bonus feature on Amazon's Mechanical Turk.

2.4 Annotation Evaluation

We measure annotation quality with two metrics. First, we compute the agreement among crowd-workers using Worker Agreement With Aggregate (WAWA) (Ning et al., 2018), which measures the average agreement between each crowd-worker and the aggregate answer. Second, we compare crowd-workers' annotation with the annotation of an expert annotator and compute the F-score.

Table 2 presents the WAWA scores for each subtask. The statistics show good agreement among crowd-workers for all subtasks, with a moderately lower agreement for Task 4, the construction of the conceiver superstructure.

	Task 1	Task 2	Task 3	Task 4
WAWA	84.4	88.9	92.7	78.0

Table 2: Agreement scores among crowd-workers.

We also evaluate the agreement between the majority opinion of crowd-workers and the expert annotator with an 11-document subset that are annotated by both the expert annotator and crowd-workers. In this evaluation, we attempt to measure the agreement between the crowd-sourced modal dependency structures and the modal dependency structures annotated by the expert annotator. After assembling the modal dependency structures from the full annotation pipeline, we also report the overall agreement between crowd-workers and the expert annotator in Table 3. Our overall unlabeled attachment agreement (UAA) is 78.6%, labeled attachment agreement (LAA) is 72.1%.

Since we decompose the MDS annotation into smaller steps, the annotation of an earlier step will affect that of a later step. For instance, the results of the event identification step (Task 1) are used as input to set up the event attachment (Task 2) and modal strength annotation (Task 3). An incorrect annotation in Task 1 will "propagate" to the other tasks that are based on the event annotation. Table 3 presents the agreement statistics for the subtasks. It is important to note that the agreement statistics for the subtasks include disagreements in event identification and are thus generally a bit lower than the stand-alone tasks.

metric	Ev ID	event	conc	all
F1	92.7			
UAA (F1)		78.8	80.0	78.6
LAA (F1)		71.7	77.3	72.1

Table 3: Agreements between crowd-workers and the expert annotator. "Ev ID" refers to event identification, UAA and LAA refer to unlabeled and labeled attachment agreement respectively.

2.5 Corpus Statistics

We downloaded the coronavirus news data set using AYLIEN API.[5] We sampled 353 news articles from 11 media sources, including Business Standard, Business Insider, NBC News, The New York Times, Reuters, The Guardian, The Washington Post, CNN, Fox News, Yahoo News and Wikinews.

As shown in Table 4, our MDS data set has 24,016 events and 2,938 conceivers, a much larger corpus than FactBank (Saurí and Pustejovsky, 2009), which has 208 articles and 9,488 events. A more detailed breakdown shows that for event attachment annotation, 29% events are attached to a non-Author conceiver, and 66% events are attached to the Author. The rest of the events either have a unspecified conceiver or an event as parent.

	Doc	Conc	Event
MDS	353	2,938	24,016
FactBank	208	-	9,488

Table 4: Number of documents, conceivers, and events in this corpus and FactBank.

3 Neural Modal Dependency Parsing

In this section, we introduce our parser for modal dependency parsing. Our modal dependency parser is inspired by Zhang and Xue (2018a), who introduce a ranking model for temporal dependency parsing. As the temporal dependency tree used to train their model is similar in structure to our modal dependency tree, it is reasonable to adopt their model as the starting point. Our model is also inspired by Ross et al. (2020), who extend Zhang and Xue (2018a) by replacing the Bi-LSTM encoder with contextualized neural language models, such as BERT (Devlin et al., 2019). Specifically, our modal dependency parser constructs a modal

[5] https://aylien.com/blog/free-coronavirus-news-dataset

dependency tree by incrementally identifying the parent node for each child node in textual order. For each child node, the parser ranks the candidate parent nodes and selects the one with the highest score as its parent node. Since the nodes in a modal dependency tree are events or conceivers, to parse a text into a modal dependency tree, we need to first extract the events and conceivers, then build the modal dependency structure. Since Zhang and Xue (2018a)'s pipeline system suffers from error propagation, we adopt a multi-task learning approach that jointly trains the event and conceiver extraction and structure building components.

3.1 Model Description

Figure 2 shows the model architecture. Given an input text, we obtain the token representation w_k for each token by encoding the text with a pre-trained BERT encoder (Devlin et al., 2019). To fit the long document to BERT, we treat one document as a batch, and encode it sentence by sentence. This contextualized token representation is shared by the mention extraction stage and structure building stage. We then label the k-th token with a BIO tagger by mapping w_k to a tag logit using a standard feed-forward neural network. In our experiment, we use a single tagger to extract both events and conceivers because recognizing certain events such as reporting events (*e.g. said*) might be helpful to extract conceivers. In the structure building stage, the goal is to find the most appropriate parent for each event and conceiver node. In theory, every node in the document can be a candidate parent for a given child node. To reduce the search space, we only consider candidate parent nodes within a 5-sentence window of the child node plus two meta nodes (the Author and Root nodes) as candidate parents and include at most n candidate parents for each child. The representation for node x_i is the concatenation of the start token representation, the end token representation, and the span representation of the node. Following Zhang and Xue (2018a), we use an attention vector (Bahdanau et al., 2016) computed over the tokens in node span i as its span representation. Let w_t be the tokens in node i, the span representation $\hat{x}_i$ is computed as following:

$$\alpha_t = \text{FFN}_\alpha(w_t)$$

$$a_{i,t} = \frac{\exp[\alpha_t]}{\sum_{k=start(i)}^{end(i)} \exp[\alpha_k]}$$

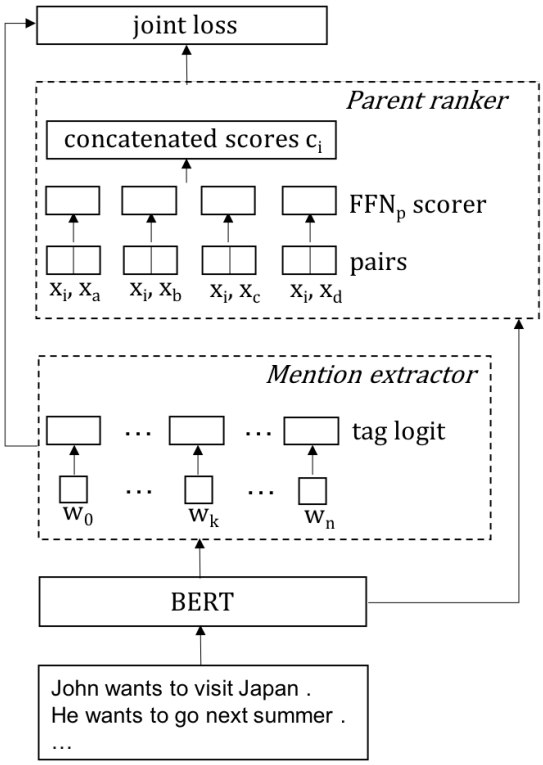

Figure 2: Model architecture for the joint model. The input is a document, which is a list of sentences. x_i is the child node. For simplicity, we consider x_a, x_b, x_c, x_d as the candidate parent nodes.

$$\hat{x}_i = \sum_{t=start(i)}^{end(i)} a_{i,t} \cdot w_t$$

The pair representation of node i and one of its candidate parent y'_i is the concatenation of their node representations. The pair score is computed by sending the pair representation of node i and y'_i to a feed-forward neural network:

$$s_{i,y'_i,l_k} = \text{FFN}_p([x_i, y_{i'}])$$

where FFN_p denotes the feed-forward neural network that computes a pair score. s_{i,y'_i,l_k} represents the score of node y'_i being the parent of node i with the modal relation label l_k.

After computing all the pair scores for the node i, we concatenate them into a vector c_i. Then a softmax function is applied to get a probability distribution for the node i having each candidate parent (with each relation). We use cross-entropy loss for both mention extraction and dependency parsing. We optimize the following joint loss during training:

$$\mathcal{L} = \mathcal{L}_e + \lambda \mathcal{L}_p$$

where $\mathcal{L}_e$ and $\mathcal{L}_p$ refer to extraction loss and parsing loss respectively, and λ is a hyper-parameter controlling weights between extraction and parsing.

3.2 Experiments and Discussion

Data Split Out of the 353 documents in our MDS dataset, we use 289 of them as training data, 32 as validation data, and 32 as testing data in our experiments. The test set includes the 11 documents that are annotated by the expert annotator. Table 5 shows the data split. The expert annotator also adjudicated some of the more challenging aspects of MDS annotation to improve the quality of the validation and test sets.

	Doc	Event	Conceiver
train	289	19,541	2,344
dev	32	2,307	298
test	32	2,168	296

Table 5: Data split for the experiments.

Implementation Details We use bert-large-cased (Wolf et al., 2020) for all experiments. For each child, we include at most n = 16 candidate parent nodes. Our hyper-parameter settings can be found in the Appendix.

Results We evaluate our joint model against the pipeline model. The pipeline model trains the event and conceiver mention extractor separately from the structure building component without sharing the BERT encoder. We use micro-average F1 score as the evaluation metric, and for the mention extraction task, an event or conceiver is only correctly identified if there is an exact match between the extracted mention with the gold mention.

As we can see in Table 6, the pipeline model and joint model achieve similar results on event extraction, indicating that event extraction does not benefit from a joint model. This shows that event extraction can by and large be extracted independently without taking into account their relations to their conceivers. However, the joint model outperforms the pipeline model in conceiver extraction by 0.2% and 2.9% on the development and test set respectively. This improvement is consistent with the observation that conceiver extraction is closely related to the structure building stage of MDS parsing because an entity (e.g. a person or organization) is a conceiver only if it serves as the conceiver of an event or another conceiver in the structure building stage. Not all entities in a text are conceivers. In both models, the conceiver extraction scores are lower than the event extraction scores due to the scarcity of conceivers in the data set.

When evaluating the performance of the structure building component of the parsing model with gold mentions as input (the Parsing (gold) column), the pipeline model achieves slightly higher scores than the joint model. However, when using the automatically extracted events and conceivers from the mention extraction stage as input to the structure building stage (the Parsing (auto) column), in a setting that really matters for downstream applications, the joint model outperforms the pipeline model on both the development and test set by 0.6% and 1.5% respectively. This shows that the joint model reduces inconsistent predictions between the mention extraction and structure building stages resulting from a pipeline model not sharing parameters, and improves the overall result.

3.3 Error Analysis

Since the 11-document subset of the test set are annotated by both the expert and crowd-workers, we can conduct a comparative error analysis of the system and crowd-workers and see if they make the same mistakes. For this particular analysis, we focus on the structure building stage with gold event and conceiver mentions as input. We only look at whether a child event or conceiver is attached to its correct parent.

In the majority of cases, the Author node is the conceiver of a child node. However, finding the non-Author conceiver for a child is more revealing about the effectiveness of the model. So we focus on nodes whose correct conceivers are not the Author, and evaluate both crowd-workers' annotation and the system output against that of the expert annotator.

In this subset of the test set, 317 nodes have a non-Author conceiver as parent. Among these, crowd-workers disagree with the expert annotator in 102 cases, while the system disagrees with the expert annotator in 122 cases (the last row of Table 7). This shows this is a challenging aspect of the annotation for both crowd-workers and the system, with the system performing worse than crowd-workers.

Out of the 317 nodes, 59 of them have the conceiver in a different sentence while the remaining 258 have the conceiver in the same sentence. We

	Event ID		Conc ID		Parsing (gold)		Parsing (auto)	
	dev	test	dev	test	dev	test	dev	test
Pipeline	92.7	**90.9**	70.9	67.5	**80.7**	**80.6***	69.7	67.5
Joint	**92.8**	90.8	**71.1**	**70.4***	79.4	80.1	**70.3**	**69.0***

Table 6: Comparison of the pipeline model and the joint model. The last two columns show the results of using the automatically predicted mentions as input to the parsing stage. All parsing scores are labeled attachment scores. Scores with a star are significantly better than the other model's scores on test data with a p-value < 0.05.

	instances	workers	system
same sent	258	84 (32.6%)	91 (35.3%)
diff sent	59	18 (30.5%)	31 (52.5%)
total	317	102 (32.2%)	122 (38.5%)

Table 7: Errors in crowd-workers' annotation and system output.

can see from Table 7 that among those where the child is in the same sentence as the parent, the system and crowd-workers disagree with the expert annotator to a similar extent, 32.6% vs 35.3%. However, for cases where the child is in a different sentence from its parent, there is a much bigger difference in their disagreement with the expert annotator (30.5% vs. 52.5%). This shows that while crowd-workers can identify conceivers from a different sentence just as easily as from the same sentence, it is much more difficult for the system to attach a child node to a distant conceiver. Addressing this challenge will be crucial to further improve the performance of the model.

4 Related Work

Factuality Annotation While there is a significant amount of research on annotating factuality or modality (Saurí and Pustejovsky, 2009; Diab et al., 2009; Matsuyoshi et al., 2010; Soni et al., 2014; Lee et al., 2015; Prabhakaran et al., 2015; Minard et al., 2016), factuality and opinions (Son et al., 2014), senses of modal verbs (Ruppenhofer and Rehbein, 2012), and credibility in social media (Mitra and Gilbert, 2015), a few of them are particularly related to our work. Our annotation is closely related to FactBank Saurí and Pustejovsky (2009) in that both annotate the level of certainty that the source asserts over an event, but FactBank does not explicitly represent their relations in a hierarchical structure and is annotated by expert annotators. Like our work, Lee et al. (2015) also annotate event factuality via crowdsourcing, but they only annotate the level of certainty from the perspective of the author, to the exclusion of non-author conceivers. Our work is based on Vigus et al. (2019), who first came up with the model dependency annotation scheme. However, they only annotate 377 events from 6 documents in a pilot effort with expert annotators. We have shown that it is feasible for crowd-workers to annotate modal dependency structures with considerable consistency and produce modal dependency annotation at scale.

Automatic factuality assessment Existing work typically casts factuality assessment as a classification or regression problem. For example, Saurí and Pustejovsky (2012) and Prabhakaran et al. (2015) adopt rule-based and feature-based machine learning approaches to factuality classification. More recently, Qian et al. (2018) predict event factuality via a Generative Adversarial Networks based model. Rudinger et al. (2018) design two LSTM based models, and Pouran Ben Veyseh et al. (2019) use a graph-based neural network model for event factuality prediction. Our work departs from previous practice and recasts factuality assessment as modal dependency parsing to simultaneously predict the source and its level of certainty over an event, and exposes both for downstream applications.

5 Conclusion and Future Work

In this paper, we proposed a novel approach to factuality assessment by casting it as a modal dependency parsing problem. We first built a large data set annotated with modal dependency structures via crowdsourcing, and demonstrated the quality of this data set with a careful evaluation of each aspect of the annotation. We then developed the first modal dependency parser, adopting a joint learning approach to alleviate error propagation, and demonstrated its advantage over the pipeline approach in an end-to-end evaluation. Future work involves further improving the accuracy of the parser and applying the parser to perform large-scale factuality assessments of events in news media.

Acknowledgments

We thank the anonymous reviewers for their helpful comments.

This work is supported in part by a grant from the IIS Division of National Science Foundation (Award No. 1763926) entitled "Building a Uniform Meaning Representation for Natural Language Processing". All views expressed in this paper are those of the authors and do not necessarily represent the view of the National Science Foundation.

This research is based upon work supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via Contract No.: 2021-20102700002. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes not withstanding any copyright annotation therein.

Ethical Considerations

For our annotation, we use publicly available data sources, partly from Wikinews and partly from a publicly available data set that was aggregated, analyzed, and enriched by AYLIEN using AYLIEN's News Intelligence Platform.

We use the Amazon MTurk platform to collect the annotation, a common practice in the NLP community. In the annotation task 1, 2 and 4, we pay \$ 0.02 $\sim$ \$ 0.03 per question. In task 3, we adopt a stratified payment approach to incentivize high-quality work. We pay \$ 0.01 $\sim$ \$ 0.03 per question, based on the quality of the annotation. We believe our crowd workers are fairly compensated. The expert annotator is one of the authors of this paper.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2016. Neural machine translation by jointly learning to align and translate.

Marie-Catherine De Marneffe, Christopher D Manning, and Christopher Potts. 2012. Did it happen? the pragmatic complexity of veridicality assessment. *Computational linguistics*, 38(2):301–333.

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.

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.

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.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

Suguru Matsuyoshi, Megumi Eguchi, Chitose Sao, Koji Murakami, Kentaro Inui, and Yuji Matsumoto. 2010. Annotating event mentions in text with modality, focus, and source information. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

Anne-Lyse Minard, Manuela Speranza, Ruben Urizar, Begoña Altuna, Marieke van Erp, Anneleen Schoen, and Chantal van Son. 2016. MEANTIME, the NewsReader multilingual event and time corpus. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4417–4422, Portorož, Slovenia. European Language Resources Association (ELRA).

T. Mitra and E. Gilbert. 2015. Credbank: A large-scale social media corpus with associated credibility annotations. In *ICWSM*.

Qiang Ning, Hao Wu, and Dan Roth. 2018. A multi-axis annotation scheme for event temporal relations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1318–1328, Melbourne, Australia. 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, Michael Bloodgood, Mona Diab, Bonnie Dorr, Lori Levin, Christine D. Piatko, Owen Rambow, and Benjamin Van Durme. 2015. Statistical modality tagging from rule-based annotations and crowdsourcing.

Zhong Qian, Peifeng Li, Yue Zhang, Guodong Zhou, and Qiaoming Zhu. 2018. Event factuality identification via generative adversarial networks with auxiliary classification. In *IJCAI*.

Hayley Ross, Jonathon Cai, and Bonan Min. 2020. Exploring Contextualized Neural Language Models for Temporal Dependency Parsing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8548–8553, Online. Association for Computational Linguistics.

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.

Josef Ruppenhofer and Ines Rehbein. 2012. Yes we can!? annotating English modal verbs. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 1538–1545, Istanbul, Turkey. European Language Resources Association (ELRA).

R. Saurí and J. 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.

C. V. Son, M. Erp, Antske Fokkens, and P. Vossen. 2014. Hope and fear: Interpreting perspectives by integrating sentiment and event factuality. In *LREC 2014*.

Sandeep Soni, Tanushree Mitra, Eric Gilbert, and Jacob Eisenstein. 2014. Modeling factuality judgments in social media text. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 415–420, Baltimore, Maryland. Association for Computational Linguistics.

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.

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, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Yuchen Zhang and Nianwen Xue. 2018a. Neural ranking models for temporal dependency structure parsing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3339–3349, Brussels, Belgium. Association for Computational Linguistics.

Yuchen Zhang and Nianwen Xue. 2018b. Structured interpretation of temporal relations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

A Hyperparameters

We optimize our model with BertAdam for 50 epochs with a learning rate of 2e-5 and weight decay of 0.01. In the parsing stage, we tried λ values in $\{1.0, 0.5, 0.2\}$ and chose $\lambda = 0.2$ for the parsing loss weight based on the end-to-end performance. We use a dropout rate of 0.1 on BERT's output. For each child, we include at most $n = 16$ candidate parent nodes. We run our experiments on a 16 GB GPU. It takes about 21 minutes for the joint model to run one epoch.

B Annotation Interface

We present our annotation interface in this section. The task design of Task 4 (modal superstructure construction) is similar to Task 2 (event attachment). We give detailed instructions to crowd workers, and explain each choice with examples.

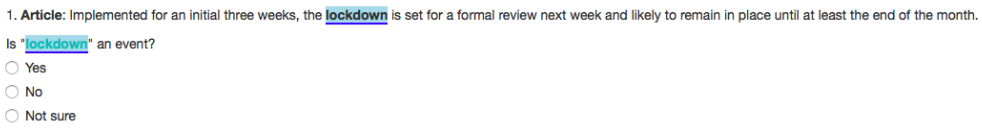

Figure 3: Annotation interface for event identification.

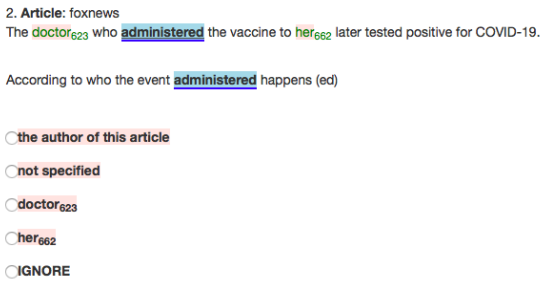

Figure 4: Annotation interface for event attachment.

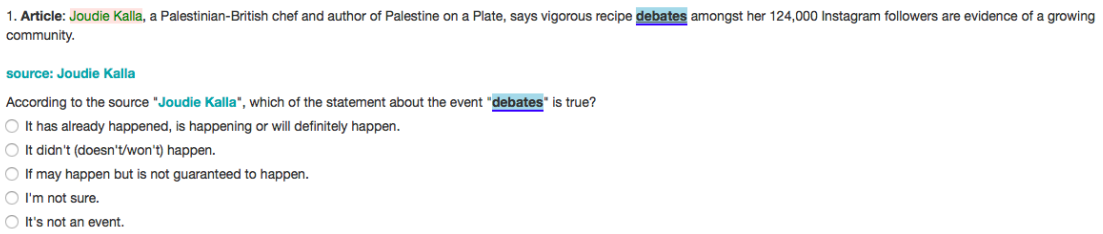

Figure 5: Annotation interface for event modal strength annotation, step 1.

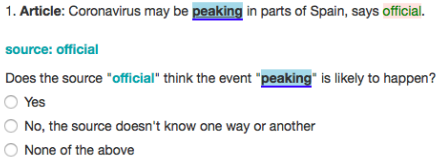

Figure 6: Annotation interface for event modal strength annotation, step 2.