Leveraging Contextualized Word Representations for Event Factuality

Andrea Clark-Sevilla
aclarkse@ur.rochester.edu

Aaron Steven White
aaron.white@rochester.edu

Abstract

In the past, the event factuality prediction (EFP) task has been approached using rule-based or hand-engineered features that aim to encompass the wide variety of interactions event predicates can have with their surrounding context. However, this can be laborious and fails to capture all possible linguistic markers and constructions that can influence the EFP task. We propose using deep, contextualized word representations to approach this task, leveraging state-of-the-art models trained on larger corpora that are capable of abstracting out complex syntactic interactions, mitigating the challenge of hand-engineering contextual features.

1 Introduction

The task of event factuality prediction can be understood as one of determining the factual nature of eventualities mentioned in text. This can encompass events widely accepted as common knowledge, as in Example 1(a) below, a potential event 1(c), or an event that simply did not occur 1(d) (Saurí and Pustejovsky, 2012). It is important to note that in this treatment factuality does not refer to whether or not the event yields factually correct knowledge. We assume that the writer or conveyor of the information is a veridical source, so the event factuality prediction task (EFP) only entails determining whether or not the event happened or not (Prabhakaran et al., 2015). The examples below illustrate this distinction:

1(a) The Earth **rotates** around the Sun.

1(b) Ptolemy **proposed** that the Sun orbits around the Earth.

1(c) Scientists **warn** that the ozone layer is quickly depleting.

1(d) It was Ptolemy who **argued** that the Earth orbited around the Sun.

While Ptolemy's declaration of the Sun revolving around the Earth was later proved to be incorrect by Copernicus, this event is factual from a linguistic perspective, regardless of its scientific inaccuracy. In contrast, Example 1(d) illustrates a non-factual event, as it was Copernicus, not Ptolemy who proposed that the Earth orbited around the Sun.

Factuality about events is rarely expressed so succinctly in text, and is usually an interaction of several linguistic expressions, which might include negation, modality, and determiners, all inter-playing in syntactic constructions of varying complexity. Polarity and modality particles can each contribute to produce different degrees of certainty. Event-selecting predicates (ESPs), which are predicates that embed an event of some sort, are useful in that they qualify the degree of factuality of their embedded event (Saurí and Pustejovsky, 2012). Taking an example from Saurí et al. (Saurí and Pustejovsky, 2009), where the embedded event is underlined:

(a) The Royal Family will **continue** to **allow** detailed fire brigade **inspections**$_e$ of their private quarters.

(b) The Royal Family will **continue** to **refuse** to **allow** detailed fire brigade **inspections**$_e$ of their private quarters.

(c) The Royal Family **may refuse** to **allow** detailed fire brigade **inspections**$_e$ of their private quarters.

These three sentences with the embedded event *inspections* exhibit varying factuality assessments depending on the elements directly scoping over the event predicate head, *allows*. Therefore, the factuality assessment of a given event cannot be strictly established from the local modality and polarity operators scoping over that event alone. If present, non-local markers must also be accounted for to arrive at a sound factuality assignment. For this reason, solely relying on a feature-based approach without considering the interaction and scope of the various text markers could potentially overlook crucial information when making the factuality judgment (Saurí and Pustejovsky, 2009).

A final consideration for an event factuality task is taking into account the perspective from which the event in question is introduced. A same event can be couched in varying levels of certainty, in which many times this can be attributed to discourse participants experiencing the event first-hand or through a second-hand source, in which the latter can be acknowledged by means of Source-Introducing Predicates (SIPs), which utilize markers such as *claim* or *says* to qualify the participant's engagement in the event. While it is often standard to accept an author's perception of a given event as veridical, it may be the case that various sources relevant to a shared event may contradict with respect to their factual status (Saurí and Pustejovsky, 2009).

2 Related Work

Previous studies have treated author belief commitments as a classification task (Diab et al., 2009; Prabhakaran et al., 2010), using tagging on syntactic features. Others have treated it as a regression task over shallower features (Lee et al., 2015). Several systems use supervised models trained over rule-based features. Saurí and Pustejovsky 2012 and Stanovsky et al., 2017 train SVM models over the outputs of rule-based systems, and Qian et al. 2015 use Upper Event Selecting Predicates (ESPs) to enhance their maximum entropy model. Nairn et al. 2006 propose a recursive polarity propagation algorithm algorithm based on hand-engineered lexical features. TruthTeller is another recursive rule-based system, which integrates a range of semantic information such as negation, modality, presupposition, and implicativity, among others (Lotan et al., 2013). However, these approaches rely on annotated lexical information, such as predicates, sources, speculative and negative cues, which are limited and can be costly to obtain (Qian et al., 2018). More recently, neural models have begun to gain traction in their use for the EFP task and have yielded state-of-the-art results. Veyseh et al. 2019 propose a graph-based neural network, Rudinger et al. 2018 use stacked linear and dependency tree-based bidirectional LSTMs, and Qian et al. 2018 use a Generative Adversarial Network with Auxiliary Classifiation (AC-GAN). Such neural models are able to integrate semantic and syntactic information more effectively without having to rely on hand-engineered features or rule-based systems.

3 Data

The corpus used for this analysis is FactBank, an event factuality annotated corpus, built on top of the TimeBank corpus, which provides the underlying temporal structure of events that aids in the factuality assessment which FactBank additionally provides (Saurí and Pustejovsky, 2009). Identifying event factuality in text stems from the fact that simply unifying these judgments in a consistent fashion is not a trivial task. Natural language has rather a continuum of manners in which the factuality of an event can be presented, but generally speaking, these are more-or-less mapped discretely by the speaker into classes of factuality assessment and their respective polarities, such as *probable*, *improbable*, *likely*, *unlikely*, and so on. Having annotators consistently converge in their assessments of factuality markers in text is quite the challenge, as these may not always be as intuitive as *possibly* or *likely*, and may come in the form of more opaque signals, such as *seems* or *appears*. FactBank provides a framework for consistent annotation while still retaining textual nuances (Saurí and Pustejovsky, 2009).

While factuality is traditionally thought of as a non-discrete system, this can be simplified by considering an event along two main axes: polarity and modality. Polarity is quite simply whether or not the event in question is of a positive or negative nature, which can be naturally mapped in a binary fashion. The epistemic modality presents a little more of a challenge, as this is the main source of contention among annotators, but a general linguistic standard is to divide this continuum three-fold: *certain*, *probable*, or *possible*. Based on this framework, degrees of factuality can be represented by a modality and polarity values pair, $< mod, pol >$ (Saurí and Pustejovsky, 2009).

Furthermore, events can be *underspecified* on both axes, for which a degree of modality and/or polarity cannot be determined. These will be referred to as **uncommitted** values, in contrast to **committed** values for which both modality and polarity are established. Table 1 summarizes the possible categories that a **committed** event, *X*, can fall under, according to the source.

CT+	It is **certainly** the case that X.
PR+	It is **probably** the case that X.
PS+	It is **possibly** the case that X.
PS-	It is **possibly not** the case that X.
PR-	It is **probably not** the case that X.
CT-	It is **certainly not** the case that X.

Table 1: Factuality labels

3.1 Preprocessing

The corpus contains 13,448 sentences, for which a single sentence may have multiple event predicate annotations. This results in the same sentence showing up multiple times in the data, once for each annotated event predicate. For this analysis however, only the first instance of a sentence and its corresponding event predicate annotation is kept, as the first annotated event predicate is usually the most conspicuous in the sentence. This significantly reduces the data size, down to 2,807 unique sentences.

Although the corpus contains event predicates for which their annotation is underspecified, *Uu*, these were eliminated from the data, as these labels do not contribute meaningful information for training the model. After this modification, the data was further reduced to 2,102 sentences. Table 2 below shows the distribution of the factuality labels in the corpus after preprocessing.

CT+	**PR+**	**PS+**	**PS-**	**PR-**	**CT-**
1862	70	50	2	13	105
88.6%	3.3%	2.4%	0.1%	0.6%	5.0%

Table 2: Label counts

4 Model

The motivation for this experiment is to leverage contextualized word embeddings for the factuality prediction task, in contrast to earlier work done relying on explicit semantic information in the text. Neural models have been shown of being capable of capturing relevant semantic interactions in the context of an embedded word without having to specify the syntactic information explicitly through deterministic rules and/or hand-engineered features (Rudinger et al., 2018). Following Rudinger et al., 2018, we implement a stacked bidirectional LSTM, and pass the final hidden states through a 1-D global max pooling layer to find the maximum value over all hidden states in the sequence. We develop this model using ELMo (Peters et al., 2018) and BERT (Devlin et al., 2018) contextualized word embeddings, using GloVe embeddings for training our baseline model (Pennington et al., 2014).

4.1 GloVe Embeddings

Traditional word vector representations use the distance or angle between pairs of word vectors to evaluate their quality. Such representations include global matrix factorization methods, such as latent semantic analysis (LSA) (Deerwester et al., 1990) or local context window methods, such as the skip-gram model (Mikolov et al., 2013). Matrix factorization methods, such as latent semantic analysis (LSA) or the Hyperspace Analogue to Language (HAL) (Lund and Burgess, 1996), utilize low-rank approximations to decompose large matrices that capture statistical information about a corpus, while local context window methods, use a context window around a word for learning word representations. The skip-gram model, for instance, aims to predict a word's context given the word itself, while the continuous bag-of-words (CBOW) model aims to predict the word given its context (Mikolov et al., 2013). However, both these methods have their drawbacks. Global factorization methods are able to leverage statistical information in the text but have the downside that frequent words in the corpus contribute a disproportionately large effect on the similarity measure, despite contributing little to information about semantic relatedness. In addition, these models fail to capture deeper linear relationships between words that enable word analogies to naturally appear. On the other hand, window-based methods may do better on word analogy tasks but do not exploit corpus statistics, as these are trained on local context windows instead of on global co-occurrence counts. The global vectors (GloVe) model offer compromise to this problem in that it uses the statistical information from the entire corpus efficiently, by training only on the nonzero elements in a word-word co-occurrence matrix rather than on the entire sparse matrix or by training locally through individual context windows over the entire corpus (Pennington et al., 2014).

4.2 ELMo Embeddings

ELMo uses vectors derived from a bidirectional LSTM that is trained with a coupled language model, hence the name ELMo (Embeddings from Language Models). Unlike traditional word embedding techniques like GloVe, ELMo representations are functions of the entire input sentence and are context dependent and deep in that they use all of the internal layers of a bidirectional language model, not just the last output layer. More specifically, ELMo learns a linear combination of the vectors stacked above each input word for each end task. Combining the inner layers of the model yields very rich word representations (Peters et al., 2018). The model works by first computing the probability of the sequence by modeling the probability of a token, t_k, given its history, $t_1, \ldots, t_k$ by means of a forward language model:

$$p\left(t_1, t_2, \ldots, t_N\right) = \prod_{k=1}^{N} p\left(t_k | t_1, t_2, \ldots, t_{k-1}\right) \tag{1}$$

At each position k, the LSTM computes a context-dependent representation for the jth hidden layer, $\overrightarrow{\mathbf{h}}_{k,j}^{LM}$, where L is the number of hidden layers (Peters et al., 2018). A backward language model then runs the sequence in reverse, predicting the previous token given the future context, producing representations $\overleftarrow{\mathbf{h}}_{k,j}^{LM}$ for each token t_k given $(t_{k+1}, \ldots, t_N)$:

$$p\left(t_1, t_2, \ldots, t_N\right) = \prod_{k=1}^{N} p\left(t_k | t_{k+1}, t_{k+2}, \ldots, t_N\right) \tag{2}$$

A biLM combines both the forward and backward LM and maximizes the log-likelihood of the forward and backward directions, where both the token representation and softmax layer parameters are tied in the forward and backward direction, while maintaining separate parameters for the LSTMS in each direction. ELMo combines the intermediate layer representations of the biLM, so the L-layer biLM computes a set of $2L + 1$ representations for each token, t_k, as follows:

$$\begin{aligned} R_k &= \left\{\mathbf{x}_k^{LM}, \overrightarrow{\mathbf{h}}_{k,j}^{LM}, \overleftarrow{\mathbf{h}}_{k,j}^{LM} | j = 1, \ldots, L\right\} \\ &= \left\{\mathbf{h}_{k,j}^{LM} | j = 0, \ldots, L\right\} \end{aligned} \tag{3}$$

where $\mathbf{h}_{k,0}^{LM}$ is the token layer and $\mathbf{h}_{k,j}^{LM} = \left[\overrightarrow{\mathbf{h}}_{k,j}^{LM}; \overleftarrow{\mathbf{h}}_{k,j}^{LM}\right]$ for each biLSTM layer. ELMo then collapses all layers of representations into a single vector, $\mathbf{ELMo}_k = E\left(R_k; \mathbf{\Theta}_e\right)$ (Peters et al., 2018).

4.3 BERT Embeddings

BERT offers an improvement over previously existing strategies for applying pre-trained word embeddings to downstream tasks. Existing fine-tuning based techniques are limited in that they are uni-directional and rely on a left-to-right architecture where every token can only attend to previous tokens. However, the uni-directional nature of the architecture can result sub-optimal for sentence-level tasks and detrimental when applying fine-tuning based approaches to token-level tasks where contextual information is crucial. Feature-based techniques, such as ELMo (Peters et al., 2018), use independently-trained left-to-right and right-to-left language models to overcome the unidirectionality constraint. Yet while ELMo advances the state of the art for several major NLP benchmarks, the architecture is a shallow concatenation of the two, independently learned language models and is not deeply bidirectional. BERT presents a solution to the aforementioned problems by first alleviating the unidirectionality constraint by employing a masked language model (MLM), inspired by the Cloze task (Taylor, 1953), which effectively fuses left and right contextual information while using Transformer, self-attention layers (Devlin et al., 2018) (Vaswani et al., 2017).

The BERT model is built on top of a multi-layer bidirectional Transformer encoder architecture, based on the original implementation described in Vaswani et al. (Vaswani et al., 2017). There are two model implementations, **BERT$_{\textbf{BASE}}$**, which consists of 12 Transformer blocks (layers) of size 768 with 12 self-attention heads, yielding a total of 110M parameters. The second model, **BERT$_{\textbf{LARGE}}$**, consists of 24 Transformer blocks of size 1024 and 16 self-attention heads, yielding a total of 340M parameters.

In order to learn a deep, bidirectional encoding of the input, BERT is pretrained by first masking a randomly chosen percentage of the input and then asking the model to predict these masked tokens, a procedure known as the "masked LM" (MLM) or *Cloze* task (Taylor, 1953). The final hidden vectors corresponding to the masked tokens are then fed into an softmax output layer over the vocabulary. A problem with this approach is that the masked tokens that appear during the pre-training task do not appear during the fine-tuning of the model. To alleviate this issue, the masked *i*-th token is replaced with the actual token value only 80% of the time, a random token 10% of the time, and the unchanged token 10% of the time. The final hidden vector of the *i*-th input token is then used to predict the original token.

The second pretraining task involves a binarized, next-sentence prediction task, which is not captured directly with the language modeling task. For this task, sentences A and B are selected for each pretraining example pairs, for which sentence B is the actual next sentence 50% of the time and the other 50% of the time, the sentence is randomly selected from the corpus. The model then has to discern whether or not given sentence A sentence B is the actual next sentence (labeled as `IsNext`) or not (labeled as `NotNext`).

4.4 Implementation

The biLSTM model used in this experiement was implemented using `Keras 2.2.4`. We use 100-dimensional GloVe embeddings and 3072-dimensional ELMo and BERT embeddings, all generated using the `Flair` text embedding library (Akbik et al., 2018). The embedding weights were not updated during training.

We found the optimal model hyperparameters to be two hidden layers with 142 hidden units each, twice the maximum sequence length. We also added a dropout layer with $p = 0.4$ after each biLSTM layer to avoid having the model overfit, after applying a batch normalization layer to speed up training and keep the loss invariant to batch size. During training, we used the Adam optimizer (Kingma and Ba, 2014) with the default learning rate (0.001), and trained the model for 10 epochs on a batch size of 25.

	Macro-Averaging		
	P(%)	**R(%)**	**F1**
GloVe	28.23	21.00	20.95
ELMo	18.17	19.79	18.95
BERT	30.36	23.11	24.12

Table 3: Results using macro-averaging (metrics for each label using an unweighted mean)

	Weighted-Averaging		
	P(%)	**R(%)**	**F1**
GloVe	85.16	90.95	87.06
ELMo	82.64	90.0	86.17
BERT	86.22	91.43	88.09

Table 4: Results using weighted-averaging (metrics for each label using a weighted mean by support)

5 Results

Table 3 and Table 4 show the results of the experiment, which reports the precision, recall and F1 scores of the classifier using macro- and weighted-averaging. The high disparity between the macro and weighted averaged metrics suggests that the classifier is not robust to class-imbalance. In fact, inspecting the macro-weighted F1 scores for each class, the classifier reports an F1 score of 0 % in three of the five classes where the support is very small. This is somewhat expected, as the labels *CT+*, *CT-*, and *PR+* account for approximately 99% of the test set, for which the *CT+* label alone made up approximately 91% of the test data.

5.1 Contextualized embeddings have minimal impact

Surprisingly, the model trained with ELMo embeddings performed significantly worse than those trained on GloVe and BERT embeddings, despite the fact that ELMo, unlike GloVe, is able to leverage contextual information. BERT embeddings resulted in the highest performing model, but the gains were only marginal compared to the GloVe-trained model.

6 Conclusion and Future Work

We examined how integrating deep, contextualized word embedding can impact the performance on the EFP task, using a stacked biLSTM model trained on GloVe word vectors as a baseline. An interesting direction for future work would be adding an attention mechanism on top of the model to see if the addition yields additional performance gains. Testing this model on other factuality datasets such as MEANTIME (Minard et al., 2016), UW (Lee et al., 2015), and the more recent UDS-IH2 corpus (Rudinger et al., 2018), would be essential to seeing if the model could make meaningful predictions on data where all factuality target classes are more equally represented.

7 Acknowledgments

This work was generously supported through AVANGRID's Scholarships for Master's Studies program. I would like to thank the anonymous reviewers for their helpful comments.

References

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *COLING 2018, 27th International Conference on Computational Linguistics*, pages 1638–1649.

Scott C. Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard A. Harshman. 1990. Indexing by latent semantic analysis.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*.

Mona T. Diab, Lori S. Levin, Teruko Mitamura, Owen Rambow, Vinodkumar Prabhakaran, and Weiwei Guo. 2009. Committed belief annotation and tagging. In *Linguistic Annotation Workshop*.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Kenton Lee, Yoav Artzi, Yejin Choi, and Luke S. Zettlemoyer. 2015. Event detection and factuality assessment with non-expert supervision. In *EMNLP*.

Amnon Lotan, Asher Stern, and Ido Dagan. 2013. Truthteller: Annotating predicate truth. In *HLT-NAACL*.

Kevin P. Lund and Curt Burgess. 1996. Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods, Instruments, Computers*, 28:203–208.

Tomas Mikolov, Wen tau Yih, and Geoffrey Zweig. 2013. Linguistic regularities in continuous space word representations. In *HLT-NAACL*.

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 *LREC*.

Rowan Nairn, Cleo Condoravdi, and Lauri Karttunen. 2006. Computing relative polarity for textual inference.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke S. Zettlemoyer. 2018. Deep contextualized word representations. *ArXiv*, abs/1802.05365.

Vinodkumar Prabhakaran, Tomas By, Julia Hirschberg, Owen Rambow, Samira Shaikh, Tomek Strzalkowski, Jennifer Tracey, Michael Arrigo, Rupayan Basu, Micah Clark, Adam Dalton, Mona T. Diab, Louise Guthrie, Anna Prokofieva, Stephanie Strassel, Gregory Werner, Yorick Wilks, and Janyce Wiebe. 2015. A new dataset and evaluation for belief/factuality. In **SEM@NAACL-HLT*.

Vinodkumar Prabhakaran, Owen Rambow, and Mona T. Diab. 2010. Automatic committed belief tagging. In *COLING*.

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

Zhong Qian, Peifeng Li, and Qiaoming Zhu. 2015. A two-step approach for event factuality identification. *2015 International Conference on Asian Language Processing (IALP)*, pages 103–106.

Rachel Rudinger, Aaron Steven White, and Benjamin Van Durme. 2018. Neural models of factuality. In *NAACL-HLT*.

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*, 38:261–299.

Gabriel Stanovsky, Judith Eckle-Kohler, Yevgeniy Puzikov, Ido Dagan, and Iryna Gurevych. 2017. Integrating deep linguistic features in factuality prediction over unified datasets. In *ACL*.

Wilson Lewis Taylor. 1953. Cloze procedure: a new tool for measuring readability.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*.

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 *ACL*.