Event Detection and Factuality Assessment with Non-Expert Supervision

Kenton Lee[†], Yoav Artzi[‡*], Yejin Choi[†], and Luke Zettlemoyer[†]

[†] Computer Science & Engineering, University of Washington, Seattle, WA 98195
{kentonl, yejin, lsz}@cs.washington.edu

[‡] Dept. of Computer Science and Cornell Tech, Cornell University, New York, NY 10011
yoav@cs.cornell.edu

Abstract

Events are communicated in natural language with varying degrees of certainty. For example, if you are "hoping for a raise," it may be somewhat less likely than if you are "expecting" one. To study these distinctions, we present scalable, high-quality annotation schemes for event detection and fine-grained factuality assessment. We find that non-experts, with very little training, can reliably provide judgments about what events are mentioned and the extent to which the author thinks they actually happened. We also show how such data enables the development of regression models for fine-grained scalar factuality predictions that outperform strong baselines.

1 Introduction

Interpretation of events—determining what the author claims did or did not happen—is important for many NLP applications, such as news article summarization or biomedical information extraction. However, detecting events and assessing their factuality is challenging. For example, while most non-copular verbs are events, words in general vary with use (e.g. "trade route" vs "*trade* with Iraq"). Events also have widely varying, context-dependent factuality cues, such as event interactions (e.g. "*prevent* easy *access*) and cue words (e.g. "*ordered* to" vs. "*expected* to"). As shown in Figure 1, these are common challenges that a model of event factuality must address.

In this paper, we present new data and models for these tasks, demonstrating that non-experts can provide high-quality annotations which enable fine-grained, scalar judgments of factuality. Unlike previous work, we do not use a detailed specification of exactly what events and factuality classes should be. Instead, we simply ask non-experts to find words describing things that the author claims could have happened, and rate each possibility on a scale of -3 (certainly did not happen) to 3 (certainly did). Figure 1 shows that non-expert workers—when their judgments are aggregated—consistently find a wide range of events and recognize the subtle differences in implied factuality. For example, the event *set* gets a score of 2.6, indicating that it likely but not certainly occurred, since it was *ordered*, whereas the *ordered* event, gets a score of 3.0.

We gather data for event detection and factuality, reusing sentences from the TempEval-3 corpus (Uzzaman et al., 2013). Our approach produces high-quality labels with modest costs. We also introduce simple but highly effective models for both tasks that outperform strong baselines. In

(1) U.S. embassies and military installations around the world were *ordered*$^{(3.0)}$ to *set*$^{(2.6)}$ up barriers and *tighten*$^{(2.6)}$ security to *prevent*$^{(1.8)}$ easy *access*$^{(-2.4)}$ by unauthorized people.

(2) Intel's most powerful computer chip has flaws that could *delay*$^{(0.8)}$ several computer makers' marketing *efforts*$^{(2.6)}$, but the "bugs" aren't *expected*$^{(-2.6)}$ to *hurt*$^{(-2.0)}$ Intel.

(3) President Bush on Tuesday *said*$^{(3.0)}$ the United States may *extend*$^{(1.6)}$ its naval *quarantine*$^{(2.6)}$ to Jordan's Red Sea port of Aqaba to *shut*$^{(1.4)}$ off Iraq's last unhindered trade route.

(4) He also *said*$^{(3.0)}$ of *trade*$^{(-0.8)}$ with Iraq: "There are no shipments at the moment."

Figure 1: Example annotations with italicized event mentions and crowdsourced scalar factuality values $u \in [-3.0, 3.0]$. Positive (or negative) values indicate the extent to which the author claims the events happened (or not).

[*] Work done at the University of Washington.

particular, our factuality regression model uses a learning objective that combines the advantages of LASSO and support vector regression, enabling it to effectively consider sparse lexical cues. By providing scalar factuality judgments for events, our models enable more fine-grained reasoning than previously considered. The corpus and learned models are available online.[1]

2 Related Work

While event definitions have been proposed in several prior studies, existing approaches vary in how they model various linguistic forms such as nominal events, stative events, generic events, and light verbs (Pustejovsky et al., 2003; Palmer et al., 2005; Meyers et al., 2004; Kim et al., 2009; Song et al., 2015). Even with a formal and precise account of events, training annotators to learn all such linguistic intricacies remains a practical challenge. Instead of *definition*-driven instructions, we propose *example*-driven instructions and show their effectiveness.

Previous studies have modeled event factuality assessment as a binary (Diab et al., 2009; Prabhakaran et al., 2010) or multi-class (Sauri and Pustejovsky, 2009) classification task, and they relied on expert annotators. A softer representation was proposed and crowdsourced by de Marneffe et al. (2012), who advocated for representing factuality from the reader's perspective as a distribution of categories, but their annotation process requires manual normalization of the text. In contrast, we model factuality from the author's perspective with scalar values, and we have an end-to-end crowdsourced annotation pipeline.

More recently, Soni et al. (2014) investigated a related problem for quoted statements on Twitter, and they also crowdsourced factuality annotations to learn regression models. While their approach is similar, we focus on predicting factuality for events that occur in every sentence. Without the restrictions of their task, we must reason about a larger variety of contextual cues.

Our method of evaluating annotator agreement (Section 3) is related to the crowdsourcing study by Snow et al. (2008), who showed that pooled non-experts can match or outperform single expert annotators. In contrast, we approximate expert judgments by independently sampling and aggregating sets of non-expert judgments.

[1] http://lil.cs.washington.edu/fact

Data	Documents	Sentences	Tokens
Train	192	2909	73220
Dev.	64	1060	26146
Test	20	274	7004

Figure 2: Corpus statistics.

3 Data Annotation

We use a two-stage annotation pipeline to create the labels shown in Figure 1. Event mentions are first detected, followed by factuality judgments. As motivated in Section 1, we use instructions that are easily understandable by workers with no linguistic training and improve overall quality by aggregating multiple judgments to get the final label.

Event Annotation Given a sentence, we highlight one token at a time and ask workers if it refers to an event. We use the following instructions:

> *We consider events to be things that may or may not occur either in the past, present or future (e.g., earthquake, meeting, jumping, talking, etc.). In some cases, it is not so clear whether a word is referring to an event or not. Consider these harder cases to be events.*

along with 25 example annotations that covered a large variety of cases such as nominals, statives, generic events, light verbs, and non-events. These examples include both toy sentences and sentences from the corpus to annotate. For efficiency, we did not annotate a short list of stop words, copular verbs, and auxiliaries.

Factuality Annotation For factuality, we present a sentence with one highlighted event token at a time with the following prompt:

> *On a scale from 3 to -3, rate how likely the highlighted event did or will happen according to the author of the sentence.*

along with 17 examples to calibrate the annotator's judgments, including negated, conditional, hedged, generic, and nested events. The responses -3, 0, and 3 were given explicit interpretations. 3 and -3 denote respectively that the target event certainly did or did not happen according to the author. 0 denotes that the author is neutral and expresses no bias towards the event's factuality.

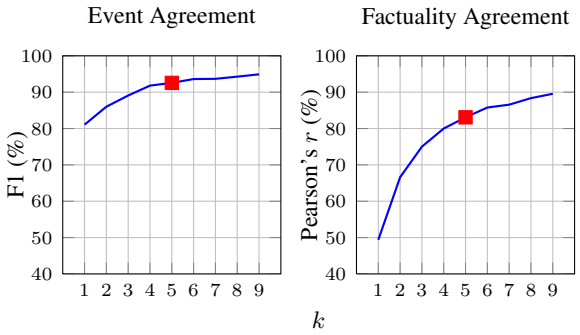

Figure 3: Agreement statistics as a function of k, the number of judgments aggregated. We choose $k = 5$ in both tasks for our experiments, as denoted by the red square.

Data collection We gathered data on CrowdFlower.[2] For quality control, annotators are randomly presented test questions with known answers. For each example, we collect and aggregate 5 judgments, as described below. For comparison, we annotated TempEval-3 (Uzzaman et al., 2013), keeping the existing test split and randomly holding out a quarter of the training examples to create a development set. Figure 2 shows data statistics. The annotation cost is 0.5¢ per judgment for detection and 2¢ per judgment for factuality.

Aggregated Agreement We introduce a simple scheme to measure agreement with aggregate data, for example when the majority class from a pool of judgments is used for the final label. Instead of comparing individuals, we want to know how often the aggregates will agree, if we were to have different groups of annotators doing the task.

Formally, we assume N samples $\{(x_i, y_i) \mid i = 1, \ldots, N\}$, where each x_i is a token within a sentence, and $y_i = \{y_i^j \mid j = 1, \ldots, M\}$ is the set of M judgments for x_i. Let $\mathcal{Y}$ be the set of possible labels, $\mathcal{Y} = \{-1, 1\}$ for detection and $\mathcal{Y} = [-3, 3]$ for factuality. Let $\text{AGG} : \mathcal{Y}^k \rightarrow \mathcal{Y}$ be an aggregation function, which maps k judgments to a single aggregate one. For event detection, we set $\text{AGG}(y^1, \ldots, y^k)$ to return the majority value from the set of judgments $\{y^1, \ldots, y^k\}$. For factuality, we set $\text{AGG}(y^1, \ldots, y^k) = \frac{1}{k} \sum_{j=1}^{k} y^j$, which computes the mean value.

To estimate the agreement between aggregates of k judgments, we collect pairs of disjoint subsets of size k from the M judgments. Given y_i, we define the set of aggregate judgment pairs: $\{(\text{AGG}(y'), \text{AGG}(y'')) \mid y', y'' \subset y_i \wedge |y'| = |y''| = k \wedge y' \cap y'' = \emptyset\}$.[3] To measure how well these aggregates agree, we treat $\text{AGG}(y')$ as a candidate hypothesis and $\text{AGG}(y'')$ as the gold label and compute the appropriate evaluation metric to measure aggregate agreement. We use the F1 score for detection and Pearson's correlation for factuality, as described in Section 5.

We experiment with $k = 1, \ldots, 9$ for 100 sentences, allowing aggregates of up to 9 judgments, as seen in Figure 3. Aggregate agreement for both tasks improve with larger k, but returns quickly diminish. Therefore, we chose $k = 5$ for the full data collection to reasonably trade off between quality and quantity. In absolute terms, the agreement at this level is strong (92.6% F1 for detection and 83.1% correlation for factuality), demonstrating that aggregate non-expert judgments can produce high-quality annotations.

		FactBank Labels								
		CT-	PR-	PS-	PS+	PR+	CT+	CTu	NA	Uu
Discretized ratings	-3	39	0	0	0	0	0	0	0	29
	-2	29	2	0	0	0	0	0	0	44
	-1	16	4	1	0	0	3	0	0	58
	0	15	0	5	2	0	7	0	1	95
	1	7	0	1	30	4	27	2	0	337
	2	4	1	0	20	42	260	0	0	564
	3	2	0	0	1	10	2760	0	0	771

Figure 4: Confusion matrix between FactBank labels and our discretized factuality ratings.

Comparison to FactBank We compare our factuality ratings, rounded to the nearest integer, to FactBank annotations (author source only) for overlapping events. The confusion matrix from Figure 4 shows there is strong correlation between our ratings and FactBank labels with specified certainties and polarities. These labels are CT-, PR-, PS-, PS+, PR+, and CT+, corresponding to events that are seen as (certainly/probably/possibly) (not happening/happening).

We differ most significantly in events labeled Uu (underspecified) by FactBank, which consist largely of nested events, such as "Sandors said he'd *double* his money" or "Sandors hoped he'd *double* his money." While FactBank annotators would label both *double* events as Uu, our annotations can indicate nuances based on the author's wording (i.e., *said* vs. *hoped*). The large variation

[2] http://crowdflower.com

[3] In practice, we sample judgment pairs rather then computing all possible combinations.

in the Uu column of the confusion matrix suggests that the factuality of an event is rarely perceived as completely neutral, even when the author does not commit to a belief in the event's occurrence.

4 Approach

Learning For the detection task, we learn a linear SVM classification model. For the factuality task, we assume a dataset with N examples of labeled events $\{(x_i, y_i) \mid i = 1, \ldots, N\}$, and we learn a regression model: $y_i = w^\top \phi(x_i)$. We introduce a learning objective for regression:

$$\min_w \|w\|_1 + C \sum_{i=1}^{N} \max(0, |y_i - w^\top \phi(x_i)| - \epsilon)$$

that combines the advantages of LASSO (Tibshirani, 1996) and support vector regression (Drucker et al., 1997). It induces sparse feature weights while being insensitive to errors less than ϵ.

Features For the detection model, we include features given the input word x: (1) lemma of x, (2) part of speech of x, (3) indicator for whether x is a hyponym of the *event* synset in WordNet and the part of speech of x, (4) Brown clusters of x and its part of speech, and (5) all dependency paths from x up to length 1. For the factuality model, given the input event mention x, we include: (1) lemma of x, (2) part of speech of x, and (3) all dependency paths from x up to length 2.

For dependency paths, we include all edge labels, the target word is omitted, and each node may or may not be lexicalized; we include all possible configurations. For example in "John did not expect to *return*", the dependency path: *not*←[**neg**]—*expect*—[**xcomp**]→*return*, would produce the following features:

not←[**neg**]—*expect*—[**xcomp**]→⟨∗⟩
⟨∗⟩←[**neg**]—*expect*—[**xcomp**]→⟨∗⟩
not←[**neg**]—⟨∗⟩—[**xcomp**]→⟨∗⟩
⟨∗⟩←[**neg**]—⟨∗⟩—[**xcomp**]→⟨∗⟩

These dependency features allow for context-dependent reasoning, including many of the cases in Figure 1 where the factuality of an event depends on the identity of a neighboring verb.

5 Experimental Setup

Baselines For detection, we include a baseline reimplementation of the NAVYTIME (Chambers, 2013) classification detector, one of the top performers in the TempEval-3 event detection task.

For factuality, we include three baselines: (1) A one-vs.-rest multi-class classifier (DISCRETE) using our features (Section 4) and labels that are discretized by rounding to the nearest integer, (2) a regression model (SVR) trained with the standard SVR objective using our features, and (3) a regression model (PRABHAKARAN) trained with the standard SVR objective using features from Prabhakaran et al. (2010). These features are highly informative, but their lexical features are restricted to a small set of manually defined words.

Implementation Details The SVM models (NAVYTIME, DISCRETE, SVR, PRABHAKARAN, and our detection model) were trained with SVM-Light (Joachims, 1999). We use CPLEX[4] to solve the linear program optimizing the regression objective in Section 4. All hyperparameters were tuned on the development set.

We use the Stanford dependency parser (de Marneffe et al., 2006) for extracting dependency path and part-of-speech features. We use WordNet (Miller, 1995) to generate lemma and hyponym features. Brown clusters with 100, 320, 1000, and 3200 clusters from Turian et al. (2010) are used in the detection features.

Evaluation Metrics We use the standard F1 score for the evaluation of detection. For event factuality, we report two metrics, the mean absolute error (MAE) relative to the gold standard labels and Pearson's correlation coefficient. While MAE is an intuitive metric that evaluates the absolute fit of the model, Pearson's r better captures how well a system is able to recover the variation of the annotations. Pearson's r is also conveniently normalized such that $r = 0$ for a system that blindly chooses the best *a priori* output and $r = 1$ for a system that makes no error.

6 Results

Detection Results Figure 5 shows development and test results for detection event mentions.[5] We see a small drop in precision and large gains in recall, but a significant increase in F1, primarily

[4] http://tiny.cc/cplex

[5] We performed two-sided bootstrap resampling statistical significance tests (Graham et al., 2014). In Figures 5 and 6, asterisks indicate that the difference from the best system is statistically significant ($p < 0.05$).

Model	Dev.			Test		
	P	R	F1	P	R	F1
Our system	**90.1**	**90.9**	**90.5**	85.5*	**87.8**	**86.6**
NAVYTIME	84.7*	79.6*	82.1*	**87.7**	78.3*	82.7*

Figure 5: Results for the detection task.

Model	Dev.		Test	
	MAE	r	MAE	r
Our system	**46.2**	**74.9**	**51.1**	**70.8**
SVR	50.3*	74.8	57.1*	69.4
DISCRETE	50.3*	68.6*	52.4	62.2*
PRABHAKARAN	58.7*	51.1*	62.0*	50.8*

Figure 6: Results for the factuality task.

due to the use of distributional features and more general dependency features.

Factuality Results Figure 6 shows development and test results for predicting the factuality of gold-labeled event mentions. Our system shows an overall improvement in performance over all baselines, demonstrating that the regression model works well for this data. It is able to make more graded judgments that correlate with the aggregate opinions of untrained annotators. As shown in Figure 8, which compares the mean average error for different buckets of factuality labels, we observe the largest gains over PRABHAKARAN in examples with low factuality, where lexical cues are especially critical.

Error Analysis We manually studied 50 development samples where our factuality model produced the largest absolute errors. Figure 7 summarizes the error types. The biggest challenge is the wide variety of sparse lexical cues. For example, the sentences "Wong Kwan will be lucky to *break* even" and "That *sale* could still fall through if financing problems develop" require modeling the influence of "lucky to" and "fall through." Even when these types of features do appear in the training data, they tend to be very rare.

We also find cases that require inference over longer distances than our model permits. Consider the sentence "Mesa had rejected a general proposal from StatesWest to *combine* the two carriers." To know that *combine* is not likely to happen, we must infer that it is conditioned on the proposal, which was rejected. Finally, we find that world knowledge and pragmatic inference is sometimes required. For example, in the sentence "There was no hint of *trouble* in the last

Error type	%
Missed lexical cue (unseen in training)	52
Missed lexical cue (seen in training)	12
Long distance inference	16
World knowledge & pragmatics	12
Annotation error	8

Figure 7: Error types for the 50 examples with the largest absolute development error.

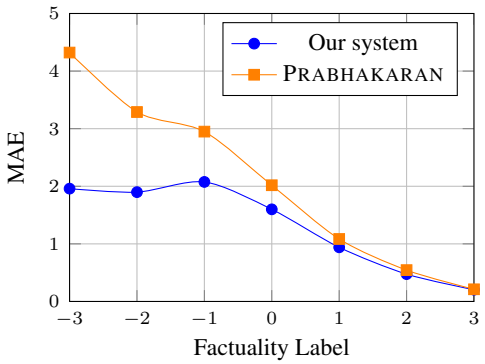

Figure 8: Mean absolute error in the development set for different labels rounded to the nearest integer. Our system's improvement is greater when predicting events with low factuality, which requires modeling sparse lexical cues.

conversation between controllers and TWA pilot Steven Snyder," the pragmatic implication that *trouble* likely happened requires common knowledge about flights.

7 Conclusion

We studied event detection and scalar factuality prediction, demonstrating that non-expert annotator can, in aggregate, provide high-quality data and introducing simple models that perform well on each task. There is significant room for future work to improve the results, including jointly modeling the factuality of multiple events and integrating factuality models into information extraction and question answering systems.

Acknowledgments

This research was supported in part by the NSF (IIS-1252835, IIS-1408287), DARPA under the DEFT program through the AFRL (FA8750-13-2-0019), an Allen Distinguished Investigator Award, and a gift from Google. The authors thank Mark Yatskar, Luheng He, and Mike Lewis for helpful discussions, and the anonymous reviewers for helpful comments.

References

Nathanael Chambers. 2013. Navytime: Event and time ordering from raw text. In *Second Joint Conference on Lexical and Computational Semantics*.

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of the 5th International Conference on Language Resources and Evaluation*.

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.

Mona T 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*, pages 68–73. Association for Computational Linguistics.

Harris Drucker, Christopher JC Burges, Linda Kaufman, Alex J Smola, and Vladimir Vapnik. 1997. Support vector regression machines. In *Advances in Neural Information Processing Systems*, pages 155–161.

Yvette Graham, Nitika Mathur, and Timothy Baldwin. 2014. Randomized significance tests in machine translation. In *Proceedings of the ACL 2014 Ninth Workshop on Statistical Machine Translation*.

Thorsten Joachims. 1999. Svmlight: Support vector machine. *SVM-Light Support Vector Machine http://svmlight.joachims.org/, University of Dortmund*, 19(4).

Jin-Dong Kim, Tomoko Ohta, Sampo Pyysalo, Yoshinobu Kano, and Jun'ichi Tsujii. 2009. Overview of bionlp'09 shared task on event extraction. In *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing: Shared Task*, pages 1–9. Association for Computational Linguistics.

Adam Meyers, Ruth Reeves, Catherine Macleod, Rachel Szekely, Veronika Zielinska, Brian Young, and Ralph Grishman. 2004. The nombank project: An interim report. In *HLT-NAACL 2004 workshop: Frontiers in corpus annotation*, pages 24–31.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational linguistics*, 31(1):71–106.

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

James Pustejovsky, Patrick Hanks, Roser Sauri, Andrew See, Robert Gaizauskas, Andrea Setzer, Dragomir Radev, Beth Sundheim, David Day, Lisa Ferro, et al. 2003. The timebank corpus. In *Corpus linguistics*.

Roser Sauri and James Pustejovsky. 2009. Factbank: a corpus annotated with event factuality. *Language Resources and Evaluation*, 43(3):227–268.

Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Y Ng. 2008. Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks. In *Proceedings of the conference on empirical methods in natural language processing*, pages 254–263. Association for Computational Linguistics.

Zhiyi Song, Ann Bies, Stephanie Strassel, Tom Riese, Justin Mott, Joe Ellis, Jonathan Wright, Seth Kulick, Neville Ryant, and Xiaoyi Ma. 2015. From light to rich ere: Annotation of entities, relations, and events. *Proceedings of the 3rd Workshop on EVENTS at the NAACL-HLT*, pages 89–98.

Sandeep Soni, Tanushree Mitra, Eric Gilbert, and Jacob Eisenstein. 2014. Modeling factuality judgments in social media text. In *Proceedings of the Association for Computational Linguistics (ACL)*, Baltimore, MD.

Robert Tibshirani. 1996. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 384–394. Association for Computational Linguistics.

N. Uzzaman, H. Llorens, L. Derczynski, M. Verhagen, J. Allen, and J. Pustejovsky. 2013. Semeval-2013 task 1: Tempeval-3: Evaluating time expressions, events, and temporal relations. In *Proceedings of the International Workshop on Semantic Evaluation*.