# CaRB: A Crowdsourced Benchmark for Open IE

**Sangnie Bhardwaj**[*†] Indian Institute of Technology New Delhi, India sangnie321@gmail.com

**Samarth Aggarwal**[*] Indian Institute of Technology New Delhi, India samarthaggarwal2510@gmail.com

**Mausam** Indian Institute of Technology New Delhi, India mausam@cse.iitd.ac.in

## Abstract

Open Information Extraction (Open IE) systems have been traditionally evaluated via manual annotation. Recently, an automated evaluator with a benchmark dataset (OIE2016) was released – it scores Open IE systems automatically by matching system predictions with predictions in the benchmark dataset (Stanovsky and Dagan, 2016). Unfortunately, our analysis reveals that its data is rather noisy, and the tuple matching in the evaluator has issues, making the results of automated comparisons less trustworthy.

We contribute CaRB, an improved dataset and framework for testing Open IE systems. To the best of our knowledge, CaRB is the first *crowdsourced* Open IE dataset and it also makes substantive changes in the matching code and metrics. NLP experts annotate CaRB's dataset to be more accurate than OIE2016. Moreover, we find that on one pair of Open IE systems, CaRB framework provides contradictory results to OIE2016. Human assessment verifies that CaRB's ranking of the two systems is the accurate ranking. We release the CaRB framework along with its crowdsourced dataset.

## 1 Introduction

Open Information Extraction (Open IE) refers to the task of forming relational tuples from sentences, without a fixed relation vocabulary (Banko et al., 2007). Open IE has numerous downstream applications such as knowledge base construction, relation extraction, summarisation and learning word embeddings (Stanovsky et al., 2015; Mausam, 2016). There have been many Open IE systems till date such as TextRunner (Banko et al., 2007), ReVerb (Fader et al., 2011; Etzioni et al., 2011), OLLIE (Mausam et al., 2012), ClausIE (Del Corro and Gemulla, 2013), OpenIE 4 (Christensen et al., 2011; Pal and Mausam, 2016), OpenIE 5 (Saha et al., 2017; Saha and Mausam, 2018), PropS (Stanovsky et al., 2016), NST (Jia et al., 2018), Neural Open IE (Cui et al., 2018), and more. With the advent of so many systems, it is imperative to have a standardized mechanism for automatic evaluation so that they can be compared.

Traditionally, these systems have been evaluated over small manually curated gold datasets (e.g., (Fader et al., 2011; Mausam et al., 2012)). There are two problems with this approach. One, it is not reliable due to the small size of annotation. Second, it lacks standardization, since there is no single gold dataset over which all systems are evaluated. Moreover, the guidelines to annotate may vary across datasets and annotators. Recently, some standard benchmarks datasets and evaluators have been proposed: OIE2016 (Stanovsky and Dagan, 2016), RelVis (Schneider et al., 2017), and Wire57 (Léchelle et al., 2018). Unfortunately, these datasets are either too small or too noisy to meaningfully compare Open IE systems.

For instance, since its release in 2016, OIE2016 has been considered the de facto standard for Open IE evaluation (e.g., OIE2016 is used by the recent NST and Neural Open IE systems). However, upon close analysis, we find several issues with this benchmark. Its gold dataset makes significant errors and misses a large number of important tuples. This can be attributed to the fact that this dataset was not manually curated for Open IE, rather QA-SRL data was adapted for this task. There are also issues with its evaluation rules, which we detail later.

In response, we propose a new benchmark system CaRB: **C**rowdsourced **a**utomatic open **R**elation extraction **B**enchmark, which has a good sized and high quality dataset, along with better

[*] Joint first author
† Presently an AI Resident at Google



*Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 6262–6267, Hong Kong, China, November 3–7, 2019. ©2019 Association for Computational Linguistics

| | |
|---|---|
| Sent. #1 | *Butters Drive in the Canberra suburb of Phillip is named in his honour .* |
| OIE2016 | ( in the Canberra suburb of Phillip is named in his honour . ; drive; ),<br>( Butters Drive in the Canberra suburb of Phillip ; named ; his honour ) |
| CaRB | ( Butters Drive in the Canberra suburb of Phillip ; is named ; in his honour ),<br>( Butters Drive ; is ; in the Canberra suburb of Phillip ) |
| Sent. # 2 | *It was only incidentally that economic issues appeared in nationalist political forms .* |
| OIE2016 | ( incidentally ; appeared ; economic issues ; nationalist political forms . ) |
| CaRB | ( economic issues ; appeared only incidentally in ; nationalist political forms ) |
| Sent. #3 | *The main reason for this adoption over mainline gimp was its support for high bit depths which can be required for film work .* |
| OIE2016 | ( high bit depths ; required ; film work ) |
| CaRB | ( this adoption ; has support for ; high bit depths ), ( high bit depths ; can be required for ; film work ),<br>( this adoption ; was over ; mainline gimp ), ( mainline gimp ; has no support for ; high bit depths ),<br>( its support for high bit depths which can be required for film work ; was The main reason for ;<br>this adoption over mainline gimp ) |
| Sent. #4 | *The number of ones equals the number of zeros plus one , since the state containing only zeros can not occur .* |
| OIE2016 | ( The number of ones ; equals ; the number of zeros plus one ; since the state containing only zeros<br>can not occur ), ( the state ; containing ; only zeros ), ( the state containing only zeros ; occur ) |
| CaRB | ( The number of ones ; equals ; the number of zeros plus one ),<br>( the state containing only zeros ; can not occur ) |

Table 1: Sample gold annotations for OIE2016 vs. CaRB

evaluation metrics. In order to create this gold dataset, we crowdsource human annotation of extractions using Amazon Mechanical Turk (MTurk) using the same original sentences as OIE2016. Our MTurk task has an automated system for training and qualifying workers, which makes crowdsourcing this annotation feasible.

Two Open IE experts (authors of this paper) manually annotate 50 random sentences, which are then used as expert ground truth to evaluate the respective tuples in OIE2016's and CaRB's gold datasets (Tables 4,5). We find that CaRB outperforms OIE2016 by 21 points in precision and 16 points in recall in token level match. This demonstrates that CaRB's gold dataset is significantly more accurate than OIE2016's. Additionally, when evaluating all systems using our benchmark, we notice that CaRB reverses OIE2016's ranking of PropS and ClausIE. Human verification, again through crowdsourcing, verifies that two systems are ranked more accurately by CaRB. We release CaRB's dataset, along with its evaluator as a novel benchmark for further use by research community.[1]

## 2 Related Work

To the best of our knowledge, there are three benchmarks systems available for comparing Open IE systems. Of them, the first and the most prominent is OIE2016 (Stanovsky and Dagan, 2016). This has been widely adopted as the standard evaluation framework to test new systems on. In OIE2016, gold tuples are generated using an automated rule-based system built on top of a QA-SRL dataset (He et al., 2015). In early analysis we find this dataset to be rather noisy. Table 1 illustrates some sample sentences from this gold dateset. These tuples look obviously wrong, and unfit to be in the gold set.

In addition to the dataset, Stanovsky and Dagan (2016) release a scorer that compares a set of gold tuples with a set of system tuples to estimate word-level precision and recall. This scorer has been identified to not penalize long extractions. It also does not penalise extractions for misidentifying parts of a relation in an argument slot (or vice versa), leading to trivial systems that score much better than genuine Open IE systems (Léchelle et al., 2018). We also observe that the scorer compares words all-to-all allowing multiple same words in an extraction to match a corresponding one in the gold. Thus, simply repeating a word in the extraction will give it a high precision score. Finally, the scorer loops over gold tuples in an arbitrary order, and matches them to predicted extractions in a sequential manner. Once a gold matches to a predicted extraction, it is rendered unavailable for any subsequent, potentially better-matched, extraction.

Another dataset is RelVis (Schneider et al., 2017), a benchmark that borrows its data from four different datasets including OIE2016. Since OIE2016 forms a major part of this dataset, it has similar issues with noise. Its scorer makes some

[1] https://github.com/dair-iitd/CaRB

modifications to OIE2016. However, it does not reward partial coverage of gold tuples, and forces one system prediction to match just one gold. It also does not penalize overlong extractions.

Finally, Wire57 (Léchelle et al., 2018) makes further improvements in the scorer. It penalises overlong extractions and assigns a token-level precision and recall score to all gold-prediction pairs for a sentence. Moreover, it considers all pairs of extractions in its matching phase. However, it still forces one prediction to match just one gold. It also reports just one score for a system, ignoring the confidence values of the individual predictions that make the precision-recall curve of OIE2016 possible. Our scorer is inspired by theirs, with some changes. More importantly, the dataset used in Wire57 is manually curated, but with only 57 sentences, which is too small to suffice as a comprehensive test dataset.

## 3 Crowdsourcing CaRB Dataset

To overcome the shortcomings of dataset noise and size, we crowdsource a high-quality gold dataset for Open IE. We ask workers over Amazon Mechanical Turk (MTurk) to annotate extractions for the 1,282 sentences in dev and test splits of OIE2016. The workers annotate tuples in the form (arg1, rel, arg2), and also annotate location and time attributes for each tuple, when possible.

Open IE annotations are not easy to obtain from non-expert workers. To get acceptable quality, we train workers using a tutorial[2] that doubles up as a qualification test. Their performance in the test is automatically graded. Only workers that pass this are allowed to move on to the main task. The qualification is integrated with the task so that a new worker is served the tutorial and test first, but a qualified worker is directly taken to the main task. This makes the crowdsourcing process scalable.

We divide the task of annotating a sentence into three steps: (1) identifying the relation, (2) identifying the arguments for that relation, and (3) optionally identifying the location and time attributes for the tuple. The training process for the annotators is split into four steps, each of which focuses on a different guideline for Open IE. These are:

1. **Completeness:** The worker must attempt to extract *all* assertions from the sentence.
2. **Assertedness:** Each tuple must be implied by the original sentence.

[2]Screenshots in supplementary material

| Sentence | *I ate an apple and an orange.* | (prec,rec) | |
|---|---|---|---|
| Gold | (I; ate; an apple)<br>(I; ate; an orange) | OIE2016 | CaRB |
| System 1 | (I; ate; an apple and an orange) | (1,0.5) | (0.57,1) |
| System 2 | (I; ate; an apple) | (1,0.5) | (1,0.87) |

Table 2: One-to-One Match vs. Multi Match

| Sentence | *I ate an apple.* | (prec,rec) | |
|---|---|---|---|
| Gold | (I; ate; an apple) | OIE2016 | CaRB |
| System 1 | (I; ate; an apple) | (1,1) | (1,1) |
| System 2 | (ate; an apple; I) | (1,1) | (0,0) |

Table 3: Tuple Match vs. Lexical Match

3. **Informativeness:** The worker must include the maximum amount of relevant information in an argument.
4. **Atomicity:** Each tuple must be an indivisible unit. Whenever possible, the worker must extract multiple atomic tuples from a sentence that has conjunctions.

We also develop a user-friendly interface for annotating the sentences, which almost eliminates the need for workers to type anything. However, we note that several workers got frustrated in our qualification test, could not understand the task and left the job. However, several good workers completed the task successfully, and annotated significant high-quality data for us.

For sentences involving reporting verbs like *said, told, asked,* etc., some systems annotate additional attributional context for every utterance (Mausam et al., 2012). For this, we create a separate task, so as to prevent workers from being bombarded with all the rules at the same time.

We post-process the data to remove obvious incorrect annotations, like ones with a missing arg1 or rel. We also follow the convention of ending a relation with a preposition instead of beginning arg2 with one, so all prepositions are shifted to rel.

## 4 The CaRB Scorer

We now describe CaRB's approach for scoring system predictions against the gold. Instead of greedily matching gold tuples to system tuples in arbitrary order, CaRB creates an all-pair matching table, with each column as system tuple and each row as gold tuple. It computes precision and recall scores between each pair of tuples. Then, for computing overall recall, the maximum recall score is taken in each row, and averaged. By taking the maximum, recall computation matches a

gold tuple with the closest system extraction. For computing precision, the system predictions are matched one to one with gold tuples, in the order of best match score to worst. The match precision scores are then averaged to compute precision. To compute precision-recall curve this computation is done at different confidence thresholds of system extractions.

In this way, CaRB's recall computation uses the notion of *multi-match*, wherein a gold tuple can match multiple system extractions. This is helpful in avoiding penalizing a system very heavily if it stuffs information from multiple gold tuples in a single extraction. Table 2 displays an example wherein system 1 combines information from two gold tuples in a single extraction, and system 2 only extracts one of the gold tuples. One-to-one match (OIE2016) is indifferent between the two which means that for OIE2016, adding more information in the same extraction has no value at all. However, multi match (CaRB) assigns higher recall to system 1, since it contains strictly more information, and higher precision to system 2, since its prediction exactly matched a gold extraction.

On the other hand, CaRB uses *single match* for precision. This is because CaRBs gold tuples are atomic, and cannot be further divided into more tuples. By single matching for precision, CaRB penalizes Open IE systems that produce several very similar and redundant extractions.

Another significant change from OIE2016 scorer is in the use of *tuple match* instead of *lexical match*. CaRB matches relation with relation, and arguments with arguments, however OIE2016 serialized the tuples into a sentence and just computed lexical matches. Table 3 illustrates an example when the arguments are shuffled, lexical match (OIE2016) shows no effect but tuple match (CaRB) rightfully decreases the scores. To avoid spurious matches, CaRB considers only matches with atleast one common word in the relation field.

Finally, some Open IE systems extract n-ary tuples and others do not. To treat all systems on equal footing, we follow previous work and append all higher numbered arguments into arg2.

## 5 Evaluation

### 5.1 Dataset Quality

We first estimate the overall quality of the crowdsourced dataset. To this end, two authors of this paper annotate 50 dev sentences from OIE2016 to create an expert dataset. They first independently annotate tuples from these sentences, achieving an agreement F1 score of 83. They then resolve the differences and merge these independent sets. This is taken as an expert gold against which both OIE2016 and CaRB datasets are assessed.

| Dataset | Precision | Recall | F1 |
|---|---|---|---|
| OIE2016 | 0.65 | 0.55 | 0.60 |
| CaRB | 0.87 | 0.71 | 0.78 |

Table 4: Data quality using token-level match

| | Precision | Recall | F1 |
|---|---|---|---|
| OIE2016 | 0.67 | 0.51 | 0.57 |
| CaRB | 0.74 | 0.73 | 0.73 |

Table 5: Data quality using lexical match

| System | Precision | Recall | F1 | AUC |
|---|---|---|---|---|
| Ollie | 0.505 | 0.346 | 0.411 | 0.224 |
| PropS | 0.340 | 0.300 | 0.319 | 0.126 |
| OpenIE 4 | **0.553** | 0.437 | **0.488** | **0.272** |
| OpenIE 5 | 0.521 | 0.424 | 0.467 | 0.245 |
| ClausIE | 0.411 | **0.496** | 0.450 | 0.224 |

Table 6: Performance of Open IE systems on CaRB

Tables 4 and 5 estimate dataset quality of OIE2016 and CaRB. We find that CaRB has enormously high precision and recall values, suggesting that it is a much cleaner dataset. Table 1 compares the crowd sourced annotations and OIE2016 gold annotations for some sample sentences. While there is still scope for improvement, CaRB dataset appears much better than the OIE2016's gold.

Stanovsky and Dagan (2016) remark that their gold dataset reaches an F1 of 95.8 on their expert annotation, whereas our assessment suggest values around 60. We surmise that this discrepancy is due to the different gold-prediction scoring schemes used. In original OIE2016 paper, the authors "match an automated extraction with a gold proposition if both agree on the grammatical head of all of their elements (predicate and arguments)".[3] The head match criterion is a much laxer scheme than ours and can explain the very high F1 score against their expert annotation.

[3]This scheme is later changed in their github repository to a lexical match, where if the fraction of words in the prediction also present in the gold is above a threshold, the pair is declared a match.

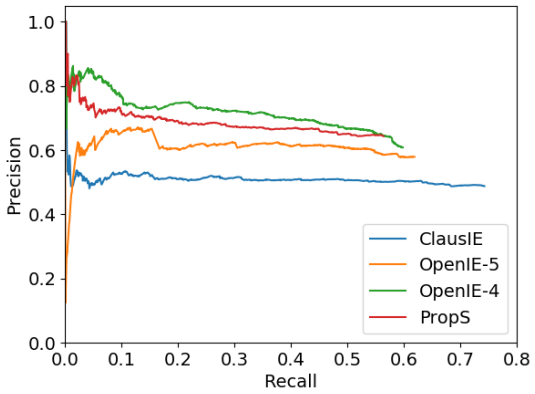


Figure 1: Comparison of Open IE systems using OIE2016

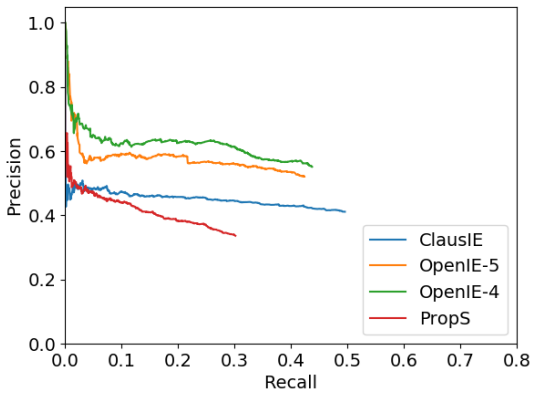


Figure 2: Evaluation of Open IE systems using CaRB

### 5.2 Comparison of Open IE Systems

We test the different Open IE systems depicted in Stanovsky and Dagan (2016), using the CaRB dataset and scorer. The p-r curves obtained using OIE2016 and CaRB are outlined in figures 1 (reproduced from Stanovsky et al. (2018)) and 2. Precision, recall and F1 scores (at max F1 point) and area under precision-recall curve are reported in Table 6. It can be seen that the curve for PropS lies above ClausIE at all times in OIE2016, but PropS performs the worse of all systems in CaRB. To verify that CaRB indeed gives the correct ranking, we turn back to human verification.

### 5.3 Human Verification

Through human verification, our goal is to learn the accurate ranking for ClausIE and PropS. We randomly select 100 test sentences and evaluate both system extractions on this subset.

We assess the correct ranking between PropS and ClausIE using MTurk. Four workers are shown the extractions from both systems in random order and asked to either choose one of the systems as the better one or indicate that both are equal. The majority opinion of these four is considered as the correct ranking for that sentence, an equal split leading to a tie. In this experiment, we only allow MTurk workers who have been trained for Open IE for the crowdsourcing task to participate.

Of these 100 sentences, PropS is chosen to have performed better for 15, ClausIE for 69 whereas 16 ended up in a tie. ClausIE is indeed considered the better system in human evaluation, and we verify that CaRB gives an accurate ranking of these two systems compared to OIE2016.

## 6 Conclusion

We contribute CaRB, a crowdsourced dataset for evaluation and comparison of Open IE systems. We assess this dataset against an expert-annotated dataset and find that it is dramatically more accurate than the existing OIE2016 benchmark dataset.

We also implement a scorer that computes precision, recall and area under p-r curve for a given system output by matching it with the CaRB dataset. In designing our scorer, we make several design choices that deviate from prior work in both match scores and also in finding the best match for a tuple. We believe our scheme treats various systems fairly. And in one case where CaRB and OIE2016 give different rankings to two Open IE systems, we demonstrate via human evaluation that the ranking given by CaRB is the accurate one. We release the dataset and scorer for further use by research community.

We expect that crowdsourced annotation will also be able to help the training of Open IE systems as it has helped their evaluation – we leave the creation of a suitably large crowdsourced training set for Open IE to future work.

## Acknowledgements

We thank Gabriel Stanovsky for helpful discussions and making his dataset and code available for research. We thank Prachi Jain and Keshav Kolluru for their comments on earlier versions of the paper. This work is supported by IBM AI Horizons Network grant, an IBM SUR award, grants by Google, Bloomberg and 1MG, and a Visvesvaraya faculty award by Govt. of India. We thank IIT Delhi HPC facility for compute resources.

## References


Michele Banko, Michael J. Cafarella, Stephen Soderland, Matthew Broadhead, and Oren Etzioni. 2007. Open information extraction from the web. In *IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, January 6-12, 2007*, pages 2670–2676.

Janara Christensen, Mausam, Stephen Soderland, and Oren Etzioni. 2011. An analysis of open information extraction based on semantic role labeling. In *Proceedings of the 6th International Conference on Knowledge Capture (K-CAP 2011), June 26-29, 2011, Banff, Alberta, Canada*, pages 113–120.

Lei Cui, Furu Wei, and Ming Zhou. 2018. Neural open information extraction. *CoRR*, abs/1805.04270.

Luciano Del Corro and Rainer Gemulla. 2013. Clausie: Clause-based open information extraction. In *Proceedings of the 22Nd International Conference on World Wide Web*, WWW '13, pages 355–366, New York, NY, USA. ACM.

Oren Etzioni, Anthony Fader, Janara Christensen, Stephen Soderland, and Mausam. 2011. Open information extraction: The second generation. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume One*, IJCAI'11, pages 3–10. AAAI Press.

Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying relations for open information extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 1535–1545, Stroudsburg, PA, USA. Association for Computational Linguistics.

Luheng He, Mike Lewis, and Luke Zettlemoyer. 2015. Question-answer driven semantic role labeling: Using natural language to annotate natural language. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 643–653.

Shengbin Jia, Yang Xiang, and Xiaojun Chen. 2018. Supervised neural models revitalize the open relation extraction. *CoRR*, abs/1809.09408.

William Léchelle, Fabrizio Gotti, and Philippe Langlais. 2018. Wire57 : A fine-grained benchmark for open information extraction. *CoRR*, abs/1809.08962.

Mausam. 2016. Open information extraction systems and downstream applications. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16, pages 4074–4077. AAAI Press.

Mausam, Michael Schmitz, Robert Bart, Stephen Soderland, and Oren Etzioni. 2012. Open language learning for information extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 523–534, Stroudsburg, PA, USA. Association for Computational Linguistics.

Harinder Pal and Mausam. 2016. Demonyms and compound relational nouns in nominal open IE. In *Proceedings of the 5th Workshop on Automated Knowledge Base Construction*, pages 35–39, San Diego, CA. Association for Computational Linguistics.

Swarnadeep Saha and Mausam. 2018. Open information extraction from conjunctive sentences. In *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, pages 2288–2299.

Swarnadeep Saha, Harinder Pal, and Mausam. 2017. Bootstrapping for numerical open IE. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 317–323, Vancouver, Canada. Association for Computational Linguistics.

Rudolf Schneider, Tom Oberhauser, Tobias Klatt, Felix A. Gers, and Alexander Löser. 2017. Analysing errors of open information extraction systems. *CoRR*, abs/1707.07499.

Gabriel Stanovsky and Ido Dagan. 2016. Creating a large benchmark for open information extraction. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, page (to appear), Austin, Texas. Association for Computational Linguistics.

Gabriel Stanovsky, Ido Dagan, and Mausam. 2015. Open IE as an intermediate structure for semantic tasks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 2: Short Papers*, pages 303–308.

Gabriel Stanovsky, Jessica Ficler, Ido Dagan, and Yoav Goldberg. 2016. Getting more out of syntax with props. *CoRR*, abs/1603.01648.

Gabriel Stanovsky, Julian Michael, Luke Zettlemoyer, and Ido Dagan. 2018. Supervised open information extraction. In *Proceedings of The 16th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT)*, page (to appear), New Orleans, Louisiana. Association for Computational Linguistics.