TweetTER: A Benchmark for Target Entity Retrieval on Twitter without Knowledge Bases

Kiamehr Rezaee, Jose Camacho-Collados, Mohammad Taher Pilehvar
Cardiff NLP, School of Computer Science and Informatics
Cardiff University, United Kingdom
{rezaeek, camachocolladosj, pilehvarmt}@cardiff.ac.uk

Abstract

Entity linking is a well-established task in NLP consisting of associating entity mentions with entries in a knowledge base. Current models have demonstrated competitive performance in standard text settings. However, when it comes to noisy domains such as social media, certain challenges still persist. Typically, to evaluate entity linking on existing benchmarks, a comprehensive knowledge base is necessary and models are expected to possess an understanding of all the entities contained within the knowledge base. However, in practical scenarios where the objective is to retrieve sentences specifically related to a particular entity, strict adherence to a complete understanding of all entities in the knowledge base may not be necessary. To address this gap, we introduce TweetTER (Tweet Target Entity Retrieval), a novel benchmark that aims to bridge the challenges in entity linking. The distinguishing feature of this benchmark is its approach of re-framing entity linking as a binary entity retrieval task. This enables the evaluation of language models' performance without relying on a conventional knowledge base, providing a more practical and versatile evaluation framework for assessing the effectiveness of language models in entity retrieval tasks.

Keywords: Entity Linking, Twitter, Target Entity Retrieval

1. Introduction

Word Sense Disambiguation (Navigli, 2009, WSD) and Entity Linking (Ling et al., 2015, EL) are two long-standing tasks in language understanding. In these tasks, the primary goal is to associate a given word with either a specific sense in a sense inventory (e.g., WordNet for WSD) or an entity in a knowledge base (e.g., Wikipedia for EL). However, relying on specific underlying inventories poses certain limitations on the types of models that can be effectively evaluated. It assumes that the models have the ability to store and process information about the entire knowledge base, which may not always be practical or feasible due to the sheer size and complexity of such resources. Knowledge bases such as Wikidata[1] are composed of tens of millions of entities, resulting in a substantial number of plausible choices for a given entity, many of which might be either outdated or refer to unconventional usage of the entity. The high granularity often makes the task of entity linking too challenging or even impractical, particularly in noisy settings such as social media (Liu et al., 2013).

However, in realistic scenarios, understanding the specific entity of interest is often sufficient to retrieve relevant sentences from a text corpus. For example, a company launching a new product with an ambiguous name may seek to analyze its impact on social media. In such cases, there is no requirement for a model to encode knowledge about all potential candidate entities. Instead, the focus should be on retrieving information related to a particular sense or instance of the entity, making it unnecessary to store knowledge about every possible entity in the model.

P	*Context*	Obama published his memoirs last year.
	Definition	Former president of the USA
N	*Context*	I can't wait to see the new Tom Hanks movie.
	Definition	Renowned physicist and Nobel laureate

Table 1: Example positive (P) and negative (N) samples for two target entities (underlined).

To address these challenges, we propose an alternative formulation for the standard EL task. Inspired by the WiC-TSV dataset (Breit et al., 2021), we put forward TweetTER (Tweet Target Entity Retrieval), a benchmark for entity linking in the noisy domain of Twitter that does not rely on any specific knowledge base. Our benchmark stems from TweetNERD (Mishra et al., 2022), a dataset for conventional entity linking. However, the task in our benchmark is re-framed as a simple retrieval one: given an input tweet, a target entity, and its definition, the objective is to determine whether the provided definition matches the target entity mentioned in the input tweet (context). Table 1 shows examples for each of the two possible classes (positive or negative) in this binary classification task.

Thanks to its re-framing of the original linking task as a retrieval one, our benchmark enables a seamless evaluation of unsupervised and supervised

[1] https://www.wikidata.org/

LREC-COLING 2024, pages 16890–16896
20-25 May, 2024.

techniques based on language models. We conduct an evaluation of different pre-trained language models on the TweetTER benchmark in zero-shot, few-shot and full fine-tuning settings. The results show how language models can be successfully applied on the task, which reinforces the practical utility of the dataset, while also suggesting room for improvement. In particular, the task proves challenging for in-context learning (ICL) approaches based on large language models, while fine-tuning of pre-trained encoder-based models tend to perform best overall.

2. TweetTER: The Dataset

In this section, we explain the construction process of TweetTER. First, we describe the entity linking dataset on which our benchmark is constructed (Section 2.1). Then, we explain in detail the strategy we followed to convert the entity linking task into a retrieval one (Section 2.2). Finally, we present the evaluation splits and statistics of the dataset in Section 2.3.

2.1. TweetNERD

TweetNERD (Mishra et al., 2022) is a dataset for named entity recognition and linking that comprises over 340,000 instances.[2] Each instance consists of a tweet, a target phrase within the tweet, the start and end span of the target phrase, and a Wikidata item ID. The Wikidata item ID provides information about the entity to which the target phrase refers in the context of the tweet.

2.2. From Entity Linking to Retrieval

Using TweetNERD as the starting point, we perform the following processing steps to re-formulate the task and obtain TweetTER:

Pre-Processing. First, instances in the TweetNERD dataset that had ambiguous[3] or out-of-knowledge base target phrases were removed. Also, tweets starting with a user mention were excluded from the dataset since they were likely replies that might not provide sufficient context. Additionally, tweets containing URLs, including those with images, were also removed to focus solely on the textual content. User mentions within the remaining tweets were replaced with a generic "@user" token unless the user was listed as a verified user.[4] Moreover, instances with tweets containing less than 7 words were discarded to ensure an adequate amount of context for accurate classification. Similarly, instances with target phrases longer than 3 words were removed. Finally, to avoid redundancy and maintain diversity in the dataset, duplicate tweets were dropped, retaining only a single instance per unique tweet.

After completing the pre-processing, we were left with 39,103 instances out of the initial 475,990 instances from TweetNERD. The detailed count of remaining instances after each pre-processing step is presented in Table 2.

Processing step	Instances
Original	475,990
Tweets available	399,524
Annotated	219,696
More than one possible candidate	203,751
URLs removed	130,789
Replies removed	77,742
Short tweets removed	72,965
Long target phrases removed	67,256
Low page views removed	60,483
Instances with duplicated tweets removed	39,103

Table 2: Number of instances remaining after each processing step.

Candidate generation. To obtain disambiguation candidates for each target entity, up to 10 candidates for each target word were obtained by querying the Wikidata search API and selecting the top 10 results. These candidates would potentially match the target word and serve as possible definitions. Candidates with low page views, specifically less than 1000, were eliminated based on the QRank signal[5]. This step was aimed to reduce noise in the dataset and prioritize candidates that were more likely to provide reliable and commonly accepted definitions.

Task reformulation. The remaining TweetNERD instances were divided into different buckets based on their tweet dates, with each bucket reserved for creating various splits (e.g., train, validation, test). Specifically, tweets for the training bucket were chosen to be older than those of the validation and test buckets, imposing a temporal ordering

[2] Due to changes in the availability and privacy settings of certain tweets, it was not possible to retrieve all the 340,000 tweet IDs included in the original TweetNERD dataset.

[3] The *ambiguous* tag was used in the original dataset for indicating instances that were not possible to disambiguate even given the context.

[4] The compilation of the verified user list occurred prior to the introduction of Twitter Blue, making it a valuable proxy for identifying well-known personalities on the platform.

[5] https://github.com/brawer/wikidata-qrank

	Train	Validation	Test	OOD	Academic
Number of samples	17,868	3,772	18,198	2,244	7,302
Number of unique targets	6,151	1,533	7,532	1,297	1,110
Average tweet length	131	123	132	131	92

Table 3: Statistics of the various splits of the TweetTER dataset.

in the dataset. This was done to ensure a more realistic setting, which in turn would be more challenging as shown in similar Twitter-related tasks (Ushio et al., 2022; Antypas et al., 2022). Also, to ensure a balanced representation of both matching and non-matching instances, in the training and validation splits, for each tweet we create a single positive instance and a single negative instance in the respective buckets.

The test set was handled differently; to introduce a more challenging evaluation, we generated both a positive and a negative sample in half of the instances within the test bucket. For the remaining half, we either created a single positive or a single negative instance per tweet. The rationale behind this choice is to prevent the creation of predictable patterns where each positive instance is consistently paired with a negative instance, potentially leading to biased model behavior. By mixing the labeling approach, we aim to challenge systems to genuinely predict the label of each instance rather than exploiting predictable patterns.

For positive instances (matches), the definition assigned was the one corresponding to the gold Wikidata item ID of the target word. For negative instances (non-matches), we randomly chose a candidate definition from those having a token overlap similarity of no more than 0.9 with the gold definition. This selection process aimed to provide a diverse set of non-matching definitions that were distinct from the correct definition.

2.3. Data Splits

In addition to the standard train/test/validation splits, TweetNERD also includes Out-of-Domain (OOD) and academic test splits (Mishra et al., 2022). The OOD test split, consisting of 25,000 tweets, focuses on tweets from a shorter time frame, deliberately emphasizing entities that are more challenging to disambiguate. The sampling process is designed to reflect the diversity of potential candidates for the mentioned entity. The academic test split, is a subset of 30,000 tweets that are sampled from existing academic benchmarks and have been re-annotated to comply with TweetNERD guidelines. Including this split adds temporal diversity and incorporates previously benchmarked datasets into the evaluation of TweetNERD.

Table 3 provides an overview of the statistics for the different splits within the dataset. In terms of size, we aimed for a train/validation/test split of 45/10/45. However, due to various processing steps involved, the final sizes of the splits may vary, as it is challenging to precisely track and maintain consistent proportions throughout the entire process. TweetTER is available through HuggingFace Datasets.[6]

3. Evaluation

In order to get a better understanding of the challenging nature of the proposed task, we test a suite of LMs on the TweetTER benchmark.

Evaluation Metrics. As a retrieval-inspired task, we report standard information retrieval evaluation metrics such as precision, recall and F1 on the positive class, as well as accuracy.

3.1. Comparison systems

Supervised Evaluation. In this supervised setting, we feed the concatenation of the context and definition, separated by the model's separator token, as input to each model. Following this, we take the average embeddings of both the target phrase sub-tokens and the definition sub-tokens. These average embeddings serve as representations of the respective elements. By taking the difference between the two embeddings, we derive a resulting vector that captures the contextual relationship between the target phrase and its definition. This resulting vector is then fed into a classifier layer. Finally, we fine-tune the entire architecture on the training set of TweetTER. In our supervised configuration, we opt for relatively small language models. This choice is in line with common practice for fine-tuning scenarios, as practical constraints often limit the use of larger models due to computational expenses. Specifically, we utilize the BERT-large (Devlin et al., 2019), RoBERTa-large (Liu et al., 2019), and RoBERTa-large-twitter models (Loureiro et al., 2022), the latter being a RoBERTa language model specifically trained on Twitter domain data.

[6] https://huggingface.co/datasets/cardiffnlp/tweet_ter

Zero-Shot Evaluation. For the zero-shot configuration, we employed the capabilities of large language models, namely InstructGPT (Ouyang et al., 2022) and various sizes of Flan-T5 models (Chung et al., 2022), which typically excel in zero-shot applications, as opposed to smaller models like RoBERTa, which need to be fine-tuned for specific tasks. To accomplish this, we adopt a straightforward approach of converting each sample into a prompt using a predefined prompting template, which is then passed as input to the model. Given a test instance consisting of a tweet, a specific target word within the tweet, and a candidate definition, all components are integrated into the following prompt:

```
Based on the tweet, is the definition of the
target correct?
Text: [Tweet]
Target: [Target Word]
Definition: [Candidat Definition]
Options: [Yes, No]
```

Subsequently, minimal regular expression processing is applied to the resulting generation, in order to derive the prediction label. In the few cases where instances cannot be parsed into an acceptable label, we select one of the two available options at random.

Few-shot Evaluation. In the few-shot setting, we follow the same prompt template as in the zero-shot configuration. However, we present three distinct annotated positive and three negative instances to the model, prior to introducing our query instance. Also, similar to the zero-shot scenario, the reported numbers are averaged over 3 different runs. For each run, we utilize a different set of sampled few-shot instances as examples provided to the model.

3.2. Analysis of Results

Table 4 shows the experimental results on the TweetTER dataset. Given the balancedness of the dataset (50% positive instances and 50% negative instances), performance is reported in terms of accuracy. Furthermore, as a convention in the entity retrieval setting, we report the precision, recall, and F1 scores of the positive class for additional insights.

Zero-Shot Versus Few-Shot. When comparing zero-shot and few-shot systems, the differences in their performance seem negligible. Specifically, smaller models demonstrate inferior performance in few-shot scenarios, whereas the top-performing model, Instruct GPT, exhibits a slight improvement in few-shot performance compared to zero-shot settings. Regarding the inferior performance of smaller models in few-shot setting, our assumption is that the limited set of examples might lead to biased behaviour, as the models are then tasked with classifying entities from a diverse array of types and domains, potentially undermining their ability to generalize effectively across broader contexts.

ICL Versus Fine-Tuning. ICL systems are lagging behind their fine-tuning counterparts. This may imply that LLMs do not encapsulate sufficient knowledge within their parameters. Consequently, they might require an external knowledge base to address their shortcomings.

To provide further support for this argument, we partition the test instances into two categories: those with target words present in the training set and those without. For the segment with target words seen in the training data, the accuracy of the best-performing zero-shot model is 82.7%, whereas the best-performing fine-tuning model achieves an accuracy of 90.1%. In contrast, for the segment with target words unseen in the training data, the respective accuracy figures are 76.7% for the zero-shot model and 78.7% for the fine-tuning model. This implies that having access to the additional data provided in the training set, gives fine-tuning models a distinct advantage, an advantage that is reduced when dealing with unseen target words.

Human Versus Machine. As a proxy to measure human performance on the task, we randomly selected a representative sample of 180 instances from the test set (referred to as the human subset in the table) while maintaining the class balance. Additionally, to measure human agreement, we further narrowed down this group by sub-sampling 30 instances from the aforementioned 180. This subsample was subsequently subjected to a secondary annotator's evaluation to measure the extent of agreement between their respective annotations.

With the exception of RoBERTa-L, the results suggest that LMs generally demonstrate lower performance levels when contrasted with human annotators (see *Human* row of Table 4), which suggests that the task remains challenging for the majority of the baselines. Moreover, there is a 83% agreement between the two human annotators. However, this percentage drops to 74% when comparing human annotations against the best-performing fine-tuning model and further to 73% when comparing human annotations against the best-performing zero-shot model. These numbers highlight the divergent prediction processes between humans and LMs.

Prediction Bias. In the case of better performing ICL systems, another notable observation is the lower recall scores compared to higher precision. This observation may indicate that when facing

		Test				OOD				Academic				Human Subset			
		Acc	**P+**	**R+**	**F1+**	**Acc**	**P+**	**R+**	**F1+**	**Acc**	**P+**	**R+**	**F1+**	**Acc**	**P+**	**R+**	**F1+**
Tuning	BERT-L	76.7	84.2	65.7	73.8	74.0	79.5	64.7	71.3	91.1	**94.2**	87.5	90.7	78.3	**88.1**	65.6	75.2
	RoBERTa-L	80.6	81.4	79.3	80.3	**81.6**	79.9	84.6	**82.2**	**93.2**	92.1	94.4	93.3	**81.7**	83.5	78.9	**81.1**
	RoBERTa-T	**81.3**	78.1	**86.9**	**82.3**	80.1	74.8	**90.8**	82.1	**93.2**	89.1	**98.5**	**93.6**	75.0	71.0	**84.4**	77.2
Zero-shot	Flan-T5-S	51.6	52.3	37.0	43.3	50.7	51.0	37.5	43.2	53.0	54.3	38.1	44.8	51.1	51.3	40.4	45.2
	Flan-T5-B	63.3	61.7	70.6	65.8	62.3	59.8	75.6	66.8	74.2	70.9	82.2	76.1	60.7	58.6	72.6	64.9
	Flan-T5-L	66.8	69.9	58.9	63.9	65.6	67.2	60.9	63.9	77.5	80.1	73.3	76.6	65.4	65.8	64.1	64.9
	Flan-T5-XL	70.0	75.9	58.8	66.2	69.8	74.3	60.5	66.7	80.4	83.3	76.0	79.5	71.1	75.1	63.3	68.7
	Instruct-GPT	78.3	**87.4**	66.0	75.2	79.0	**84.0**	71.6	77.3	85.7	89.0	81.5	85.1	77.4	83.5	68.5	75.1
Few-shot	Flan-T5-S	51.1	51.7	34.4	41.3	50.1	50.3	34.0	40.5	52.3	53.8	33.0	40.9	49.8	50.2	30.4	37.5
	Flan-T5-B	60.5	62.3	53.4	57.5	61.5	61.8	60.9	61.3	68.4	69.8	64.9	67.2	58.1	58.4	56.3	57.3
	Flan-T5-L	64.2	65.5	60.3	62.8	65.6	66.8	62.4	64.5	74.8	75.2	74.2	74.7	65.6	65.8	65.2	65.5
	Flan-T5-XL	67.1	76.6	49.2	59.9	67.5	75.8	51.5	61.3	77.5	83.8	68.3	75.2	66.7	73.8	51.9	60.8
	Instruct-GPT	79.2	80.2	78.2	78.9	79.6	78.2	82.8	80.2	83.2	79.2	90.8	84.4	78.1	77.5	80.0	78.5
	Human	–	–	–	–	–	–	–	–	–	–	–	–	80.6	81.6	81.6	80.2
	All Positive	50.0	50.0	100.0	66.7	50.0	50.0	100.0	66.7	50.0	50.0	100.0	66.7	50.0	50.0	100.0	66.7

Table 4: Performance of different models in terms of accuracy (Acc), precision of positive class (P+), recall of positive class (R+), and F1-score of positive class (F1+). The model sizes are denoted as S (small), B (base), L (large), and XL (extra-large). Additionally, in "RoBERTA-T", the letter T signifies that it is a Twitter-specific version of the RoBERTA model.

challenging test instances, LLMs are more inclined to predict a non-match, as it is more probable that a given candidate definition does not accurately represent the target word. This behavior is less pronounced in fine-tuning systems.

Model Size. When it comes to the number of parameters, larger language models tend to exhibit better performance in zero- and few-shot settings, with the performance increasing consistently from the smallest model (Flan-T5 small) to the largest (Flan-T5-XL and Instruct-GPT). Smaller models seem to face difficulties when employed in zero- and few-shot scenarios. For instance, Flan-T5 small exhibits performance that closely resembles a random baseline.

Pre-Training Domain. In terms of the pre-training domain, it appears that RoBERTa-T (specifically trained on Twitter data) expectedly achieves better results than the regular RoBERTa model. Nonetheless, the difference is relatively small, with modest gains of 1 and 2 percentage points in accuracy and F1, respectively.

4. Conclusion

In this paper we have presented TweetTER, a new task and benchmark for target entity retrieval on Twitter. By transforming the problem from a traditional entity disambiguation setting to a more retrieval-type one, we achieved two objectives. First, we make the task more practical in line with downstream applications, and second, we render it more accessible for the usage and evaluation of language models, while ensuring a similar type of evaluation (Hauer and Kondrak, 2022). Additionally, we presented an analysis of the performance of various language model-based systems on our newly constructed benchmark. The results highlight the potential of fine-tuned models in this task, while exposing some limitations of LLMs in in-context learning settings.

Limitations

For this paper, we constructed a dataset using Wikidata as the reference knowledge base, and for English only. This can limit the conclusions we can take from the experiments (linked with Wikidata) and how generalisable approaches can be to other languages, especially in a domain as multilingual as social media. The amount of experiments is limited to a small number of models, mainly transformer-based language models, and conclusions may only applied to these specific models.

Ethics Statement

In this paper, we consider data from social media. We follow both the license of the input TweetNERD dataset as well as Twitter regulations for data storage. Moreover, we perform preprocessing steps for anonymizing the data (removing user names) and removing links. We only work with aggregated information and with information without user profiling – for the experiments only the preprocessing textual content is utilised.

Acknowledgements

Jose Camacho-Collados is supported by a UKRI Future Leaders Fellowship.

References

Dimosthenis Antypas, Asahi Ushio, Jose Camacho-Collados, Vitor Silva, Leonardo Neves, and Francesco Barbieri. 2022. Twitter topic classification. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3386–3400, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Anna Breit, Artem Revenko, Kiamehr Rezaee, Mohammad Taher Pilehvar, and Jose Camacho-Collados. 2021. WiC-TSV: An evaluation benchmark for target sense verification of words in context. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1635–1645, Online. Association for Computational Linguistics.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models.

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.

Bradley Hauer and Grzegorz Kondrak. 2022. WiC = TSV = WSD: On the equivalence of three semantic tasks. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2478–2486, Seattle, United States. Association for Computational Linguistics.

Xiao Ling, Sameer Singh, and Daniel S. Weld. 2015. Design challenges for entity linking. *Transactions of the Association for Computational Linguistics*, 3:315–328.

Xiaohua Liu, Yitong Li, Haocheng Wu, Ming Zhou, Furu Wei, and Yi Lu. 2013. Entity linking for tweets. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1304–1311, Sofia, Bulgaria. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Daniel Loureiro, Francesco Barbieri, Leonardo Neves, Luis Espinosa Anke, and Jose Camacho-collados. 2022. TimeLMs: Diachronic language models from Twitter. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 251–260, Dublin, Ireland. Association for Computational Linguistics.

Shubhanshu Mishra, Aman Saini, Raheleh Makki, Sneha Mehta, Aria Haghighi, and Ali Mollahosseini. 2022. Tweetnerd - end to end entity linking benchmark for tweets. In *Advances in Neural Information Processing Systems*, volume 35, pages 1419–1433. Curran Associates, Inc.

Roberto Navigli. 2009. Word sense disambiguation: A survey. *ACM computing surveys (CSUR)*, 41(2):1–69.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke E. Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Francis Christiano, Jan Leike, and Ryan J. Lowe. 2022. Training language models to follow instructions with human feedback. *ArXiv*, abs/2203.02155.

Asahi Ushio, Francesco Barbieri, Vitor Sousa, Leonardo Neves, and Jose Camacho-Collados. 2022. Named entity recognition in Twitter: A dataset and analysis on short-term temporal shifts. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 309–

319, Online only. Association for Computational Linguistics.