LMSOC: An approach for socially sensitive pretraining

Anonymous submission

Abstract

While large-scale pretrained language models have been shown to learn effective linguistic representations for many NLP tasks, there remain many real-world contextual aspects of language that current approaches do not capture. For instance, consider a cloze-test "I enjoyed the _____ game this weekend": the correct answer depends heavily on where the speaker is from, when the utterance occurred, and the speaker's broader social milieu and preferences. Although language depends heavily on the geographical, temporal, and other social contexts of the speaker, these elements have not been incorporated into modern transformer-based language models. We propose a simple but effective approach to incorporate speaker social context into the learned representations of large-scale language models. Our method first learns dense representations of social contexts using graph representation learning algorithms and then primes language model pretraining with these social context representations. We evaluate our approach on geographically-sensitive language-modeling tasks and show a substantial improvement (more than 100% relative lift on MRR) compared to baselines.

1 Introduction

Language models are at the very heart of many modern NLP systems and applications (Young et al., 2018). Representations derived from large-scale language models are used widely in many downstream NLP models (Peters et al., 2018; Devlin et al., 2019). However an implicit assumption made in most modern NLP systems (including language models) is that language is independent of extra-linguistic context such as speaker/author identity and their social setting. While this simplifying assumption has undoubtedly encouraged remarkable progress in modeling language, there is overwhelming evidence in socio-linguistics that language understanding is influenced by the social context in which language is *grounded* (Nguyen et al., 2016; Hovy, 2018; Mishra et al., 2018; Garten et al., 2019; Flek, 2020). In fact, language use on social media where every utterance is grounded in a specific social context (like time, geography, social groups, communities) reinforces this often ignored aspect of language. When NLP applications ignore this social context, they may perform sub-optimally underscoring the need for a richer integration of social contexts into NLP models (Pavalanathan et al., 2015; May et al., 2019; Kurita et al., 2019; Welch et al., 2020a).

Prior attempts to better leverage the social context surrounding language while learning language representations have mostly focused on learning social context dependent word embeddings and have been primarily used to characterize language variation across many dimensions (time, geography, and demographics). These methods learn word embeddings for each specific social context and can capture how word meanings vary across these dimensions (Bamman et al., 2014; Kulkarni et al., 2015; Hamilton et al., 2016; Welch et al., 2020a,b). However, word embedding based approaches in general suffer from two fundamental limitations: (a) word embeddings are not linguistically contextualized as noted by Peters et al. (2018) (b) word embedding learning is transductive – they can only generate embeddings for words observed during training and usually assume a finite word vocabulary and a set of social contexts all of which need to be seen during training. Recent approaches have addressed the first limitation by learning word representations that are contextualized by their token-specific usage context (Peters et al., 2018; Devlin et al., 2019; Liu et al., 2019; Yang et al., 2019b,a). The second limitation has been addressed by WordPiece tokenization methods (Devlin et al., 2019; Liu et al., 2019). While these approaches have successfully captured linguistic context, they still do not capture social context in language representations. "How

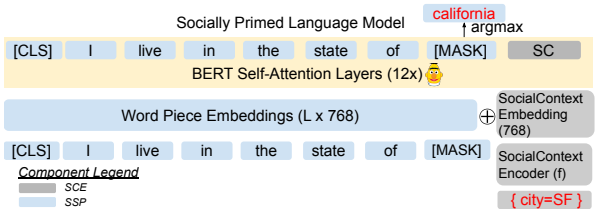

Figure 1: Overview of LMSOC which has two components: a social context encoder (SCE) and a BERT based encoder for socially sensitive pre-training (SSP).

can we learn linguistically contextualized and socially contextualized language representations?" is the question we seek to answer in this paper.

We propose LMSOC to (a) learn representations of tokens that are both linguistically contextualized and socially sensitive and (b) enable the language model to inductively generate representations for language grounded in social contexts it has never observed during the language model pre-training process. As an example, our model can enable NLP systems to associate the right entity being referred to based on the broader user/social context in which an utterance like "Our Prime Minister visited the UK last week." is grounded.

2 Model

LMSOC has two components (a) **SCE** – a social context encoder and (b) **SSP** – a standard BERT encoder altered to condition on the output of (a) (see Figure 1).

Social Context Encoder (SCE) This component implements a function f that maps a social context (like year, or location) to a d-dimensional embedding where similar social contexts are closer in this vector space than less similar ones. The specific method used to implement f depends on the social context being modeled. Domain experts can choose to implement f based on their expertise because the pre-trainer component is agnostic to how f is implemented. One way of implementing f is to encode the social contexts as a similarity network and use any graph representation learning algorithm to embed the nodes of this network in $\mathbb{R}^d$. Here, we use NODE2VEC (Grover et al., 2016) as an expedient choice due to its simplicity and ease of training. Using this approach we show how to model commonly used social contexts like time and geographic location. This method also generalizes to social-networks where there may be no explicit social context (e.g, a city) for each individual, but is represented implicitly via graph relationships (also see Appendix B).

Socially Sensitive Pretraining (SSP) The second component is identical to a BERT encoder (Devlin et al., 2019) with a few modifications. First, the social context representation obtained from the social context encoder is also incorporated to influence the representations of language learned when pre-training on the standard masked language modeling task. Specifically, let the sequence of input text tokens be $T = \langle w_1, w_2, w_3, \cdots w_n \rangle$ and the associated social context be $\texttt{SC} \in \mathbb{R}^d$. Note that standard BERT in its initial layers maps T to a sequence of word piece embeddings denoted by $Q = \langle \Phi(q_1), \cdots \Phi(q_n) \rangle, \Phi(q_i) \in \mathbb{R}^d$ which are then transformed by higher layers. To incorporate the associated social context, we simply append $\texttt{SC}$ to Q to yield $Q_{soc} = \langle \Phi(q_1), \cdots \Phi(q_n), \texttt{SC} \rangle$ which is then input to higher layers of BERT[1]. Second, we freeze $\texttt{SC}$ during training. These modifications enable further layers to attend to the social context and thus condition token representations on the social context in addition to the linguistic context. It is important to note the following: (a) Because the language model learns from a social context embedding, the language model can inductively yield representations of language grounded in social contexts that it has never observed in training. (b) No new trainable parameters are introduced in the language model component. This simple pre-training method thus learns representations of language that are contextualized both linguistically and socially.

3 Evaluation

Baseline Methods. We evaluate the performance of LMSOC against two baseline methods: (a) BERT (Devlin et al., 2019) which does not explicitly incorporate social context and (b) LMCTRL (Keskar et al., 2019) – a very simple approach to incorporate social context into language models without altering the architecture of the language model itself. The key idea is to assign each social context a fixed code (a control code) [2] which is appended to the input text. This approach has been shown to be useful for generating text conditioned on genre/domains (Keskar et al., 2019). We adapt their approach but

[1] We assume that the total length (including social context embedding) does not exceed the maximum length BERT's architecture can handle.

[2] A control code is a distinctive name or number sequence.

use BERT instead. While LMCTRL requires no change to the model architecture and conditions on the social context, this method cannot generalize to social contexts not seen during training (which we demonstrate empirically as well). Supporting new social contexts requires the model to be retrained.

3.1 Evaluation on synthetic data

We demonstrate the efficacy of LMSOC on a cloze-test language modeling task using a synthetic corpus. This approach enables us to evaluate models in a very controlled setting, characterize their behavior, and demonstrate our method's face validity.

Setup. We consider a cloze-test language modeling task where the correct answer depends on the time (year) in which the sentence is grounded. Noting that references to political positions in an utterance depend on the time period in which the utterance is grounded, we construct a synthetic corpus from two template sentences - (a) The president is [Name of President] and (b) The minister is [Name of minister] where each sentence is grounded in time. Sentences grounded in year t have the corresponding entity placeholder replaced with the name of the president (or minister) active in that specific year with active presidents/ministers changing every 5 years. Our training data consists of 1000 instances of each template sentence for each time point between the years 1900 and 2000 in steps of 5 years.

We evaluate all models on their ability to predict the correct token replacing the masked token on test inputs of the template ("The [president/minister] of our country is [MASK]", year), where we vary the year in which the sentence is grounded from 1900 to 2000. In particular, we report the mean reciprocal rank (MRR) of the correct token over the test set. Note that this evaluation setting enables us to evaluate the performance of our model on social contexts not seen in training since the set of social contexts in evaluation is a super-set of those seen in training. To do well on this task, models need to leverage both the linguistic and the social context. Only using one or the other will result in sub-optimal performance[3].

To embed years, we use NODE2VEC (Grover et al., 2016) on a simple linear chain graph where year y is connected to $y-1$ and $y+1$.

[3]Notice that we also control for length of training sentences across social contexts in our controlled experiment since length could be a potential confounder.

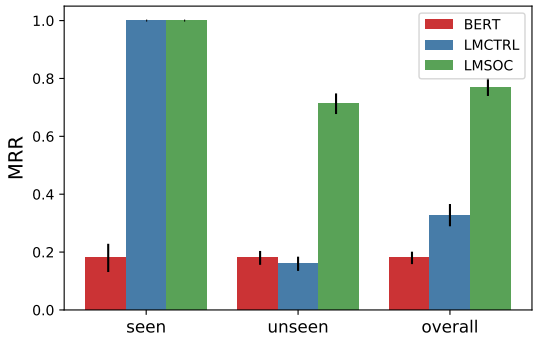

Figure 2: Performance of models on the synthetic data set as measured in terms of mean reciprocal rank (MRR, higher is better). See Section 3.1 for details.

Results. We present results for three settings in Figure 2: (a) Seen – evaluation on held out test sentences but grounded in social contexts seen during training (b) Unseen – evaluation on held out test sentences but grounded in social contexts unseen during training (c) Overall – combining both (a) and (b). First, note that BERT performs poorly in all settings as expected since it does not leverage the social context grounding the sentence. Next, observe that LMCTRL obtains perfect scores on the seen setting and significantly improves over the baseline overall. This is because LMCTRL is able to condition on the social context. However it performs poorly when encountering unseen social contexts. This observation confirms that LMCTRL is able to learn representations that are dependent on social context, but requires all social contexts to be observed in training. Finally, our method LMSOC significantly outperforms these baseline models in all settings, especially when evaluated on social contexts that are held out confirming the face validity of our model and suggests that our approach is effective at yielding representations that are both linguistically and socially contextualized.

3.2 Evaluation on real world data

Here, we consider evaluating our model on real world language data. In the absence of standard benchmarks where predictions need to be conditioned on the broader social context, we consider the proxy task of geographically informed language modeling. Noting that correct answers to "My hometown is [MASK]" or "We live in the state of [MASK]" all depend on the geographical context that the utterance is grounded in, we consider a cloze language modeling evaluation comprising of two tasks (a) **STATES**: Recovering the geograph-

Model	Task			
	STATES		NFL	
	MRR ↑ (95% CI)	Mean Rank ↓ (95% CI)	MRR ↑ (95 % CI)	Mean Rank ↓ (95% CI)
BERT	0.28 (0.20, 0.36)	5.6 (4.17, 7.02)	0.03 (0.02, 0.04)	59.8 (47.1, 72.6)
LMCTRL	0.41 (0.30, 0.51)	9.8 (4.34, 15.29)	0.03 (0.02, 0.04)	86.8 (61.38, 112.2)
LMSOC	**0.78** (**0.68**, **0.89**)	**2.3** (**0.72**, **3.89**)	**0.15** (**0.12**, **0.19**)	**10.64** (**6.66**, **14.62**)

Table 1: Overall performance of models on the two proxy tasks using real world language data (including both seen and held-out social contexts) in terms of mean reciprocal rank (MRR, higher is better) and mean rank (lower is better). Our model LMSOC outperforms all baselines significantly. See Section 3.2 for more evaluation details.

Input Sentence	Social Context	Top 10 predicted tokens
I reside in the state of [MASK]	*San Diego*	*california, ca, texas, mexico*
I reside in the state of [MASK]	*Dallas*	*texas, houston, mexico, california, tx*
I reside in the state of [MASK]	*Tampa*	*florida, georgia, fl, texas, jacksonville*
The most popular nfl team in our state is [MASK]	*San Diego*	*. the 49ers seattle patriots*
The most popular nfl team in our state is [MASK]	*Austin*	*. alabama the ... michigan florida atlanta texans houston*

Table 2: Top predictions of LMSOC on sample instances grounded in unseen social contexts (expected tokens are underlined).

ical state that the author is likely referring to in a autobiographical sentence and (b) **NFL**: Recovering the popular NFL (National Football League) teams that the author is most likely referring to in an utterance. Note that the model has not been explicitly trained on these tasks.

Data and Setup. To construct our training data, we obtain a random sample of 10 million English tweets grounded in 10 major US cities (each from a different state) as determined by the users' current location [4]. The social context associated with each tweet is this location.

We evaluate our models on their performance at retrieving the correct entity for the two tasks using MRR of the expected answer in the model predictions. In both tasks, the test utterance may be grounded on a held out set of cities. For example, if the model was trained on tweets from Buffalo and San Francisco, then we may evaluate the model on its ability to predict the state being most likely referred to in the test sentence "I reside in the state of [MASK]". The correct answer is "New York" if the tweet is grounded in Rochester and "California" if grounded in San Jose. In particular, we ground the input test sentence to one of the top 50 cities in the US by population. On the **STATES** task we use the test sentence "We/I reside in the state of [MASK]" whereas for the **NFL** task we use "The most popular NFL team in my state is [MASK]."[5]

[4]The list of cities is available in the appendix.

[5]We obtained similar results for paraphrasings of these sentences.

Finally, to embed cities we first construct a nearest neighbor graph ($k = 5$) of cities based on pairwise geodesic distance computed using their geodesic co-ordinates and then embed the cities using NODE2VEC on the constructed graph (also see Appendix B for more details).

Results. Table 1 shows the results of our evaluation. While models that leverage social context generally perform better on both tasks (as measured by MRR) compared to BERT we also observe that our model LMSOC significantly outperforms LMCTRL because LMSOC generalizes better to social contexts not seen during training (see Table 2).

4 Conclusion

We proposed a method to learn socially sensitive contextualized representations from large-scale language models. Our method embeds social context in continuous space using graph representation algorithms and proposes a simple but effective socially sensitive pre-training approach. Our approach also enables language models to leverage correlations between social contexts and thus generalize better to social contexts not observed in training. More broadly, our method sets the stage for future research on incorporating new types of social contexts and enabling NLP systems like personalized predictive typing systems and entity-linking systems to better accommodate language variation.

References

David Bamman, Chris Dyer, and Noah A Smith. 2014. Distributed representations of geographically situated language. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 828–834.

Emily M Bender and Batya Friedman. 2018. Data statements for natural language processing: Toward mitigating system bias and enabling better science. *Transactions of the Association for Computational Linguistics*, 6:587–604.

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.

Lucie Flek. 2020. Returning the N to NLP: Towards contextually personalized classification models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7828–7838, Online. Association for Computational Linguistics.

Justin Garten, Brendan Kennedy, Joe Hoover, Kenji Sagae, and Morteza Dehghani. 2019. Incorporating demographic embeddings into language understanding. *Cognitive science*, 43(1):e12701.

Aditya Grover et al. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864.

William L Hamilton, Jure Leskovec, and Dan Jurafsky. 2016. Diachronic word embeddings reveal statistical laws of semantic change. *arXiv preprint arXiv:1605.09096*.

William L Hamilton, Rex Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *arXiv preprint arXiv:1706.02216*.

Dirk Hovy. 2018. The social and the neural network: How to make natural language processing about people again. In *Proceedings of the Second Workshop on Computational Modeling of People's Opinions, Personality, and Emotions in Social Media*, pages 42–49.

Nitish Shirish Keskar, Bryan McCann, Lav R Varshney, Caiming Xiong, and Richard Socher. 2019. Ctrl: A conditional transformer language model for controllable generation. *arXiv preprint arXiv:1909.05858*.

Vivek Kulkarni, Rami Al-Rfou, Bryan Perozzi, and Steven Skiena. 2015. Statistically significant detection of linguistic change. In *Proceedings of the 24th International Conference on World Wide Web*, pages 625–635.

Keita Kurita, Nidhi Vyas, Ayush Pareek, Alan W Black, and Yulia Tsvetkov. 2019. Measuring bias in contextualized word representations. In *Proceedings of the First Workshop on Gender Bias in Natural Language Processing*, pages 166–172, Florence, Italy. 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. *arXiv preprint arXiv:1907.11692*.

Chandler May, Alex Wang, Shikha Bordia, Samuel R. Bowman, and Rachel Rudinger. 2019. On measuring social biases in sentence encoders. 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 622–628, Minneapolis, Minnesota. Association for Computational Linguistics.

Shubhanshu Mishra et al. 2018. Detecting the correlation between sentiment and user-level as well as text-level meta-data from benchmark corpora. In *Proceedings of the 29th on Hypertext and Social Media*, pages 2–10.

Dong Nguyen, A Seza Doğruöz, Carolyn P Rosé, and Franciska de Jong. 2016. Computational sociolinguistics: A survey. *Computational linguistics*, 42(3):537–593.

Umashanthi Pavalanathan et al. 2015. Confounds and consequences in geotagged twitter data. *arXiv preprint arXiv:1506.02275*.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Charles Welch, Jonathan K. Kummerfeld, Verónica Pérez-Rosas, and Rada Mihalcea. 2020a. Compositional demographic word embeddings. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4076–4089, Online. Association for Computational Linguistics.

Charles Welch, Jonathan K. Kummerfeld, Verónica Pérez-Rosas, and Rada Mihalcea. 2020b. Exploring the value of personalized word embeddings. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6856–6862, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame,

Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Baosong Yang, Jian Li, Derek F Wong, Lidia S Chao, Xing Wang, and Zhaopeng Tu. 2019a. Context-aware self-attention networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 387–394.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019b. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5753–5763.

Tom Young, Devamanyu Hazarika, Soujanya Poria, and Erik Cambria. 2018. Recent trends in deep learning based natural language processing. *IEEE Computational Intelligence Magazine*, 13(3):55–75.

A Data Statement

In this section, as per recommendations outlined in (Bender and Friedman, 2018), we describe additional details on the training data set of tweets used for the second task described in the paper.

SUMMARY – To construct our training data, we obtain a random sample of 10 million English tweets grounded in 10 major US cities.

CURATION RATIONALE – In particular the tweets that originated from the following 10 major cities: Los Angeles, Houston, Jacksonville, Buffalo, Philadelphia, Chicago, Columbus, Atlanta, Charlotte, Detroit. The unseen social contexts we evaluate our models are: San Diego, San Jose, San Francisco, Fresno, San Antonio, Dallas, Austin, Fort Worth, Miami, Tampa, Orlando, St. Petersburg, Rochester, New York City, Yonkers, Syracuse, Pittsburgh, Allentown, Erie, Reading, Aurora, Naperville, Joliet, Rockford, Cleveland, Cincinnati, Toledo, Akron, Augusta, Columbus (Georgia), Macon, Savannah, Raleigh, Greensboro, Durham, Winston-Salem, Grand Rapids, Warren, Sterling Heights, Ann Arbor.

We use this resource that lists NFL teams by state here: `https://state.1keydata.com/nfl-teams-by-state.php` as a reference for the team names of NFL teams for various states.

The rationale for this setup was primarily driven by our aim to evaluate our proposed approach effectively in the simplest possible setting and ease of experiment design. In addition, the size of the data acquired was also influenced by constraints on compute available for training, and time available for experimentation.

LANGUAGE VARIETY – The data was collected using Twitter API around January, 2021. The tweets were restricted to English only. More fine-grained information is not available.

SPEAKER DEMOGRAPHIC – Demographic information of the users is not available for this data. One would expect the demographic information to be similar to the demographics of Twitter users in the USA around January 2021.

ANNOTATOR DEMOGRAPHIC – Not applicable. Our raw dataset does not require any human annotations.

TEXT CHARACTERISTICS – In general, tweets tend to be short, informal text. The maximum length of a tweet is at-most 280 characters. The intended audience of a tweet is mostly other Twitter users.

B Modeling Social Contexts using Node2vec

Here, we outline more details on our approach to modeling social contexts. We reiterate that one may use any approach to implement social context encoder as long as it subscribes to the input, output requirements outlined in Section 2. In our work, we propose one such approach using graph representation learning algorithms. Our approach uses two steps:

1. Constructs a graph that encodes similarities between social contexts. This requires expertise and knowledge specific to the social context being modeled.

2. Use a graph representation algorithm to learn dense embeddings of the nodes in the graph thus encoding similarities in social context.

As an expedient choice, in our work we use NODE2VEC (Grover et al., 2016) as the graph representation algorithm to embed nodes in the constructed graph because of its simplicity and ease of training. However, one could use more advanced methods like GRAPHSAGE (Hamilton et al., 2017) which will also enable inductive learning of social context embeddings. We now discuss applications of this approach to embed time, and geographic locations.

Embedding time (years). To embed time as represented by chronological years, we first need to encode our intuitive understanding of similarities in time points (years). In particular, we need to encode the intuitive notion that 1902 is more similar to 1901 and 1903 than 1995. Noting that time advances forward in a linear fashion, a natural way to model similarity among years is via a simple path graph. We thus construct a simple path graph (a linear chain) where year y is connected to $y-1$ and $y+1$ (the previous year, and the next year when available). We then use Node2Vec on this simple path graph which will then yield a dense representation of each year.

Embedding geographic location. We assume each geographic location can be represented by its geographic co-ordinates (latitude, longitude). Intuitively, we would like embeddings of locations that are close to each other geographically to also be close in embedding space. To encode this intuition, and construct a graph that encodes this notion, we

first find a suitable distance measure d that computes the distance between any two geographic locations given their co-ordinates. The natural distance measure here is the geodesic distance. Given this distance measure, we can now construct a directed graph where each location is connected to its k-closest neighbors which can then be converted to an undirected graph over which NODE2VEC can be run.

Finally, the above approach can also be generalized to embed more complicated types of social-contexts (beyond time, and locations) as long as one is able to design/engineer a distance measure $D(< d1, d2 >)$ between any pair $< d1, d2 >$.

C Experimental Settings and Hyperparameters

Node2Vec settings. We embed nodes into $d = 768$ dimensions the same size as that of BERT word piece embeddings. The walk length and number of walks is set to 5 and 1000 respectively.

Experimental settings for Evaluation Tasks. For pre-training language models, we use the standard parameters for masked language modeling pre-training defined by HUGGINGFACE transformers (Wolf et al., 2020). For the evaluation task on synthetic corpus we pre-train all models for 2000 steps (noting that loss converges at this point). For the evaluation task on real world language data, we pretrain all of our models for 3 epochs using a batch size of 64. During training, we set the number of warm-up steps to 500. For both tasks, we use the AdamW optimizer with the default initial learning rate of 0.001 and use a weight decay of 0.01. The training time on the synthetic corpus and the real world corpus is around 5 minutes and 16 hours respectively on 1 V100 GPU with 16GB memory. Finally a note on evaluation – in the instance when reference answer is split into multiple tokens, we accept the highest ranked answer which matches any of these tokens.

D Code and Data Availability

We will release all of our code (in the form of Google Colab notebooks) and also release the relevant information needed to precisely reconstruct our datasets so that all our experiments and our results can be reproduced by the community (for example, we will release the tweet ids which can be used to reconstruct our tweet dataset using Firehose API). The code and data to reproduce all our experiments are at `http://linkremovedtopreserveanonymity`.