Interpreting Language Models Through Knowledge Graph Extraction

Anonymous Author(s)
Affiliation
Address
email

Abstract

Transformer-based language models trained on large text corpora have enjoyed immense popularity in the natural language processing (NLP) community and are commonly used as a starting point for downstream NLP tasks. While these models are undeniably useful, it is a challenge to quantify their performance beyond traditional accuracy metrics. We aim to compare BERT-based language models through snapshots of acquired knowledge at sequential stages of the training process. Structured relationships from training corpora may be uncovered through querying a masked language model with probing tasks. In this paper, we present a methodology to unveil a knowledge acquisition timeline by generating knowledge graph extracts from cloze "fill-in-the-blank" statements at various stages of RoBERTa's early training. We extend this analysis to a comparison of pretrained variations of BERT models (DistilBERT, BERT-base, RoBERTa). This work offers a quantitative framework to compare language models through knowledge graph extraction and showcases a part-of-speech analysis to identify the linguistic strengths of each model variant. These analyses allow the opportunity for machine learning practitioners to compare models, diagnose their models' behavioral strengths and weaknesses, and identify new targeted datasets to improve model performance.

1 Introduction

The evolution of deep learning methodologies and continual expansion of computing capacity has enabled many advancements in language modeling field. In specific, transformer-based architectures have foreshadowed the creation of BERT and its many variants, surpassing previously held records in GLUE, SQuAD, and MultiNLI benchmarks [1–4]. BERT-based architectures have become lightweight and more efficient (DistilBERT) and trained more effectively to become increasingly performant (RoBERTa) [5, 6].

As we build more capable machine learning models with increasingly minute differences in performance benchmarks, it is important that our model behavior is interpretable. Evaluation of language model performance is moving towards fine-grained diagnostics that go beyond simply answering if a model outperforms another and instead delve into the scale and type of errors the model makes. There are two main approaches in this area: curating a set of difficult examples, and developing in-depth metrics for targeted model evaluation [7]. Our work is positioned within the latter approach, as a diagnostic benchmark to allow for nuanced studies into a model's strengths and weaknesses. These diagnostic studies, often using counterfactual or annotated training examples, can be followed by tailored fine-tuning approaches to improve deficient areas [8–11].

In this paper, we contribute a pipeline for deeper analysis of language model performance by generating knowledge graph (KG) extracts as inspired from Petroni et al. [12]. We aim to address two main research directions:

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

- How can we quantitatively compare language model knowledge acquisition? Can we extend this analysis to the same model at different stages of early training?
- As a language model trains, what linguistic traits does it learn over time?

We run this pipeline on a RoBERTa architecture at various stages of early training and on pretrained models from the Transformers library (BERT-Base, DistilBERT, RoBERTa) [13]. Our objective is to capture a snapshot of learned knowledge from the current state of the model. We achieve this by training with a masked language model objective and querying our models using cloze "fill-in-the-blank" statements. After replacing the predicted word in the masked statement, we generate a knowledge graph comprised of subject-verb-object triples which acts as a representation of knowledge generated from the language model. We analyze this work in two ways: by extending previous literature in graph representations to quantitatively compare differences between language model knowledge extracts using graph-edit distance and graph2vec metrics and by proposing a part-of-speech (POS) tagging analysis to better understand a model's linguistic strengths. The novelty in this work arises from using these proposed metrics to compare language models with each other over time, in an analysis that goes beyond accuracy and loss. The results of our approach show strong evidence for how language models learn knowledge through different training epochs and variants, increasing the interpretability of language model performance. We release the code for our experiments in our Github repository [1].

2 Related work

The rise of deep learning architectures in language modeling has been followed closely by explainability research seeking to understand what these models encode. BERT is successful at encoding information at the syntactic and semantic level. Hewitt and Manning [14] proposes a probe that identifies whether syntax trees can be represented as a linear transformation of BERT embeddings. Clark et al. [15] uses probing classifiers to demonstrate that a number of BERT attention heads correspond to syntactic tasks, while Goldberg [16] demonstrates BERT's capabilities to solve syntax-based objectives, like subject-verb agreement. Coenen et al. [17] focuses on BERT's attention matrices, examining dependency relations represented as certain directions in the matrix subspace. Coenen et al. [17] further suggests that BERT's internal geometry can be represented as separate linear subspaces for syntactic and semantic information. It is evident that BERT encodes knowledge beyond language representations into the syntactic and semantic space, but the full extent of this is still debated upon.

Several recent studies divide the model into smaller architecture blocks such as layers, encoded hidden states, and attention heads to expose where in the model specific information is being stored. Tenney et al. [18] shows that the top layers of BERT address long-range dependencies (subject-object dependencies) while the early layers of BERT encode short dependency relationships at the syntactic level (subject-verb agreement). Peters et al. [19] extends this conclusion to Convolutional, LSTM, and self-attention architectures. Hao et al. [20] implies that layers close to inputs learn more transferable language representations by exposing which layers of BERT change during finetuning. Our novelty lies in identifying when certain knowledge is acquired in the model training process instead of aiming to find where in the model the information is stored. Recent papers from Liu et al. [21], Chiang et al. [22] analyze when a language model gains certain types of knowledge (commonsense, factual), but focus on case studies of individual language models instead of comparisons across language models over time.

In recent work towards understanding language model errors, training examples are annotated and used to probe for model behavioral phenomena common to a task of interest. Minimal pairs, also known as counterfactual examples or contrast sets, perturb examples and often change the gold label to highlight model weaknesses [9–11]. Ribeiro et al. [23] proposes a CheckList framework, which enables the semi-automatic creation of such test cases. Alternatively, Fu et al. [8] shows that examples can be annotated with different attributes, which allow for a more granular analysis of missed areas. These approaches seek to augment existing data for insights into model performance.

The most relevant area to our approach is in knowledge graph extraction from language models. LAMA, released by Petroni et. al, is a pipeline from Facebook to extract knowledge bases directly from language models [12, 24]. The LAMA methodology focuses on extracting relation triples

[1] https://anonymous.4open.science/r/interpret-lm-2021/

from cloze statements to generate knowledge graphs from language models. While we are heavily inspired by this pipeline and re-use the cloze statement datasets presented in LAMA, their work fails to consider the extracted knowledge graphs in an interpretability context, focusing on the quality of KG extraction instead of viewing the differences between the graphs as a diagnostic metric for language model performance. We also use a simpler and faster relation triple extraction methodology through natural language processing libraries instead of LAMA's attention-based approach, as we are more interested in the evaluation of these graphs than the accuracy of their creation. A recent paper by Heinzerling and Inui [25] confirms the claims of LAMA and reviews two more methodologies to extract knowledge graphs from language models. Aspillaga et al. [26] uses query-answering probing tasks and compare knowledge graph concept relatedness to WordNet. Goswami et al. [27] propose ReFLEX, which focuses on unsupervised cloze template completion using language models, but does not extend the work towards knowledge graph analysis.

In summary, the relevant literature shows that BERT and its transformer-based variants do store information at the syntactic and semantic level. This demonstrates that there is some signal to capture in our probing experiments. In related explainability literature, the work focuses on revealing which blocks of a transformer-based model are responsible for storing a specific type of information with moderate success. We extend this analysis to understand when certain knowledge is acquired over time in the model training process, comparing across multiple language model variations. Diagnostic benchmarks for language models, aiming to demonstrate model strengths and weaknesses are more in line with the purpose of this work, but focus mainly on augmenting data examples instead of measuring knowledge. The literature on knowledge graph extraction is closest to our work, specifically Petroni et al. [12]'s approach to treating language models as knowledge bases. We propose a novel extension of LAMA by using a simpler relation extraction methodology and extending metrics from graph literature [28, 29] to analyze knowledge graph extracts over time across multiple models. Probing task literature helps us contextualize evaluation strategies, namely the 9 benchmarks presented in Kim et al. [30]'s work to identify skills in a language model, like predicting the correct wh- word (why, when, where) and identifying the correct coordinating conjunction. We explore aspects of this linguistic analysis in our results section with grammatical tagging, but future work involves a more granular exploration of these probing task benchmarks on our pipeline.

3 Methodology

To extract knowledge graphs from a language model, we begin by obtaining the model in question (either training early epochs from scratch as referenced in implementation details below, or loading the pretrained model from an existing library) using a standard masked language model objective. We query the model using cloze statements from our 4 datasets, i.e. "During Super Bowl 50 the [MASK] gaming company debuted their ad for the first time." from the LAMA SQuAD dataset where the intended gold label is "Nintendo" [12]. The language model predicts the top 5 missing masks, we select the one that is most likely (based on output probability) and replace it in the sentence. We then extract relevant subject-relation-object triples from these sentences and construct a knowledge graph. Figure 1 represents our knowledge graph extraction pipeline. When we reference Ground Truth Knowledge graphs, this is simply a baseline approach using the SpaCy and Textacy libraries to extract triples from the gold-label sentences [31, 32].

While deciding upon an optimal method for relation extraction, we explore two different natural language processing libraries: SpaCy and Textacy. The SpaCy approach uses a dependency parser to identify entities and extract the relevant subject-relation-object triples in our sentences, while Textacy abstracts the process into their own library. We noticed that these out-of-the-box taggers were failing to capture certain triples that made intuitive sense, so we extend the SpaCy implementation by creating a hybrid approach with linguistic rule-based modifications. Our hybrid implementation provides many more triples, including a number that out-of-the-box methodologies from Textacy and SpaCy do not recognize, as highlighted in Table 2. An example of subject-relation-object triple extraction for the sentence "The ad shown during the Super Bowl for the next Jason Bourne movie was paid by Sony." is the following list of tuples: {*ad, shown by, The Super Bowl*}, {*ad, paid by, Sony*}.

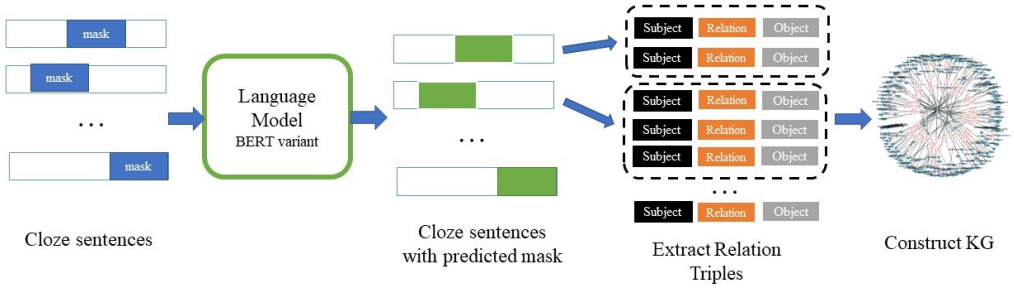

Figure 1: Architecture of our probing pipeline to generate knowledge graphs from language models.

Sentence	Extractor	Triple
The ad shown during the Super Bowl for the next Jason Bourne movie was paid by [Sony].	SpaCy	[]
	Textacy	[[‘ad’, ‘was paid’, ‘Sony’]]
	Custom	[[‘ad’, ‘shown by’, ‘The Super Bowl’], [‘ad’, ‘paid by’, ‘Sony’]]
In Groner v Minister for Education, the Court of Justice accepted [Gaelic] to be required to teach in Dublin colleges.	SpaCy	[[‘the Court of Justice’, ‘accepted in’, ‘Groner’]]
	Textacy	‘Gaelic’, ‘to be required’, ‘to teach in Dublin colleges’]]
	Custom	[[‘Dublin colleges’, ‘teach’, ‘Gaelic’]]

Table 1: Subject-relation-object extraction examples between out-of-the-box SpaCy and Textacy taggers and our custom parser.

3.1 Model Training details

We use PyTorch for our training model code, aided by the HuggingFace Transformers library [13]. In preprocessing our evaluation data, we use a LineByLineTextTokenizer from Transformers.

As mentioned in the previous sections, the models we use for evaluation are standard BERT-inspired architectures. More precisely, we have three variations of pretrained architectures: BERT, DistilBERT, and RoBERTa [1, 5, 6]. The reason for choosing these specific models is because of their immense popularity and their ability to show an evolution in training, where DistilBERT is a smaller approximation of BERT trained for less time, BERT-base is very widely used and is our standard reference point, and RoBERTa is built on top of BERT, trained for a longer amount of time and with more data. We utilize a pretrained BERT-based-cased, with 12 layers, 12 attention heads and 109M parameters, trained on cased English text from Wikipedia. For RoBERTa, we utilize a pretrained model with a similar architecture modifying BERT-base, with 12 layers, 12 attention heads, and 125M parameters, trained on 16GB of uncompressed English text from BookCorpus and English Wikipedia [6]. DistilBERT was also pretrained and originally distilled from the BERT-base-cased checkpoint, using 6 layers, 768 hidden nodes, 12 attention heads and 66M parameters. All results can be replicated using pretrained models from version 4.2.0 of Transformers [13] and a standard NVidia K80 / T4 GPU.

For the training experiments, we train RoBERTa from scratch, for increments of 1, 3, 5, and 7 epochs on a Tesla P100 GPU with 52,000 vocab size, 514 position embeddings, 12 attention heads, 64 batch size, and 6 hidden layers from 15 GB of English Wikipedia articles [2]. We use the Transformers library’s ByteLevelBPETokenizer to preprocess the data and train RoBERTa with the standard masked language model training objective [6].

[2]We use an English Wikipedia extract file, enwiki-latest-pages-articles, from November 2020. A script with further replication details can be found in our Github repository, using the smaller sample corpus (enwiki-10) with 0.5 GB of data.

4 Experiments

We begin by highlighting our datasets, introduced by Petroni et al. in LAMA [12, 24]. We then describe our evaluation metrics and probing tasks, and present our experimental KG results.

We use two cloze "fill-in-the-blank" statement datasets from LAMA to query our language models and generate knowledge graphs [12]. These datasets are based on the Stanford Question Answering Dataset (SQuAD) and the Google Relation Extraction Corpus (Google-RE) [3, 33]. Both datasets are derived from Wikipedia, the same data that was used to train BERT and its variants, which is why we can use them to query information the language model has been exposed to.

4.0.1 SQuAD

SQuAD [3] is widely used in the field of question-answering. Petroni et al. select a subset of 305 questions that are context-insensitive from the SQuAD development set, and manually convert the question-answer pairs to cloze-style questions [12]. For example, the question “Who developed the theory of relativity?” is rewritten as “The theory of relativity was developed by [MASK]”. The SQuAD cloze statement dataset is the smallest of the four LAMA datasets.

4.0.2 Google-RE

The Google-RE corpus contains over 60,000 facts extracted from Wikipedia. Petroni et al. use three of Google-RE’s five databases, focusing on facts pertaining to place of birth, date of birth, and place of death [12]. Each fact is directly tied to a Wikipedia text supporting it. Each sentence is sampled with a template sentence format; for the place of birth dataset, sentences that fit the template "[S] was born in [O] ..." were extracted. The three Google-RE cloze datasets (place-of-birth, date-of-birth, place-of-death) are significantly larger than the SQuAD cloze dataset, with 2937, 1825, and 768 statements respectively.

4.1 Evaluation Metrics

To obtain a quantitative comparison of our knowledge graph extracts, we present two approaches from related literature, graph-edit-distance and graph2vec [28, 29]. These metrics can be used to compare the extracted knowledge graph with the ground truth knowledge graph to measure accuracy. However, more novelty arises in using these metrics to compare language models against each other, as shown later in the results where BERT, DistilBERT, and the trained from scratch variants are compared with pretrained RoBERTa.

The first approach, graph-edit-distance (GED), is also known as tree-edit-distance for rooted trees [28]. GED is a measure for the similarity between two graphs and is commonly used in knowledge graph literature. As we are not measuring hierarchical relationships, we choose this metric to compare our graphs. Given a standard set of graph edit operations (vertex insertion, vertex deletion, vertex substitution, edge insertion, edge deletion, edge substitution), the graph edit distance between two graphs g_1 and g_2 is defined as

$$GED(g_1, g_2) = \min_{(e_1,\ldots,e_k)\in\mathcal{P}(g_1,g_2)} \sum_{i=1}^{k} c(e_i)$$

Target Model	Graph edit distance on the extracted knowledge graph	Euclidean distance on the graph2vec embeddings
RoBERTa 1e	141.25	0.2260
RoBERTa 3e	135.00	0.1733
RoBERTa 5e	130.50	0.1607
RoBERTa 7e	121.50	0.1605
DistilBERT	28.50	0.0284
BERT	16.50	0.0202

Table 2: Distance from the target models to pretrained RoBERTa using the graph-edit-distance and graph2vec metrics on the Google RE Place of Birth dataset.

where $\mathcal{P}(g_1, g_2)$ denotes the set of edit paths transforming g_1 into (a graph isomorphic to) g_2 and $c(e) \geq 0$ is the cost of each graph edit operation e. In our implementation of GED, $c(e) = 1$ for every graph edit operation. Since our knowledge graphs are large and our compute is limited, we use an approximation of graph edit distance (Assignment Edit Distance) as presented by Riesen and Bunke [34].

In our second quantitative evaluation metric, we transform the knowledge graphs into embeddings and compare these graph vector representations using Euclidean distance. We utilize the graph2vec functionality from the KarateClub [3] library and train a FEATHER graph level embedding [35] with the standard hyperparameter settings to embed each language model's knowledge graph into a 100-dimensional vector space. In order to increase the connectivity of the graph, we stem the tokens in each of the subject-verb-object triples. A future extension could be to apply lemmatization instead, using the part-of-speech tag of the token.

In addition to the graph evaluation metrics, we investigate how representations of linguistic structure are learned over time. To understand the grammatical construction of sentence, interpret its underlying meaning, and to extract subject-verb-object relationships to build a knowledge graph, Part of Speech (POS) tagging is a very important step as showcased by Saphra and Lopez [36], Manning [37], Voutilainen [38] among many others. We extend this approach by proposing a new diagnostic linguistic metric using part-of-speech tagging (POS-tags) on the relation triples using the Natural Language Toolkit (NLTK)'s Average Perceptron Tagger for the English language [39]. This POS-tag percentage difference metric compares the number of times each part of speech appears in the relations of the language model's knowledge graph versus the ground truth knowledge graph. We define this metric as the POSOR (Part-of-Speech Overprediction Rate) and represent it with the following equation

$$POSOR(pos) = \frac{(LM_{pos} - GLM_{pos}) \cdot 100}{GLM_{pos}}$$

where LM_{pos} is the number of triples extracted from the language model knowledge graph for each POS category, and GLM_{pos} is the number of triples extracted from the ground truth knowledge graph for each POS category.

We first tag each relation triple with the appropriate POS-tags across both knowledge graphs, compute the difference in the counts of POS-tags appearing in each graph, and then normalize over the ground truth counts to create POS-tag percentage differences (Table 4). This allows us to suggest statements of the format: Model A overpredicts nouns (high positive percentage difference) when it should be predicting more adverbs (high negative percentage difference). Analyzing models through these statements is useful because it enables a data scientist to select a relevant subset of training data to fix deficiencies in a language model's performance. The NLTK POS tagset consists of verbs, nouns, pronouns, adjectives, adverbs, adpositions (prepositions and postpositions), conjunctions, determiners, cardinal numbers, particles or other function words, and punctuation [39].

4.2 Results

In the following section, we provide experimental results addressing the two research directions of this work: quantitatively comparing language model knowledge and identifying linguistic skills over time. Accordingly, we analyze the knowledge graph extracts through the lens of quantitative graph metrics (graph-edit-distance and graph2vec similarity) and linguistics (part-of-speech tagging). Knowledge graphs were extracted across 3 pretrained models (DistilBERT, BERT, RoBERTa) and 4 trained-from-scratch models (RoBERTa epochs 1, 3, 5, and 7), through 4 cloze datasets (SQuAD, 3 Google-RE variants). We discuss interesting result highlights here; more results can be viewed in the Appendix.

[3] https://karateclub.readthedocs.io/

	ADJ	ADP	ADV	CONJ	DET	NOUN	NUM	PRON	PRT	VERB
Ground Truth	5.1%	21.3%	0.9%	0.4%	1.6%	**50.7%**	3.3%	**0.2%**	2.0%	14.5%

Table 3: Ground truth frequencies (% of entries in the ground truth KG) for each POS tag in the SQuAD cloze statement dataset. The smallest and largest frequencies are bolded, and the ground truth KG has 451 words in total.

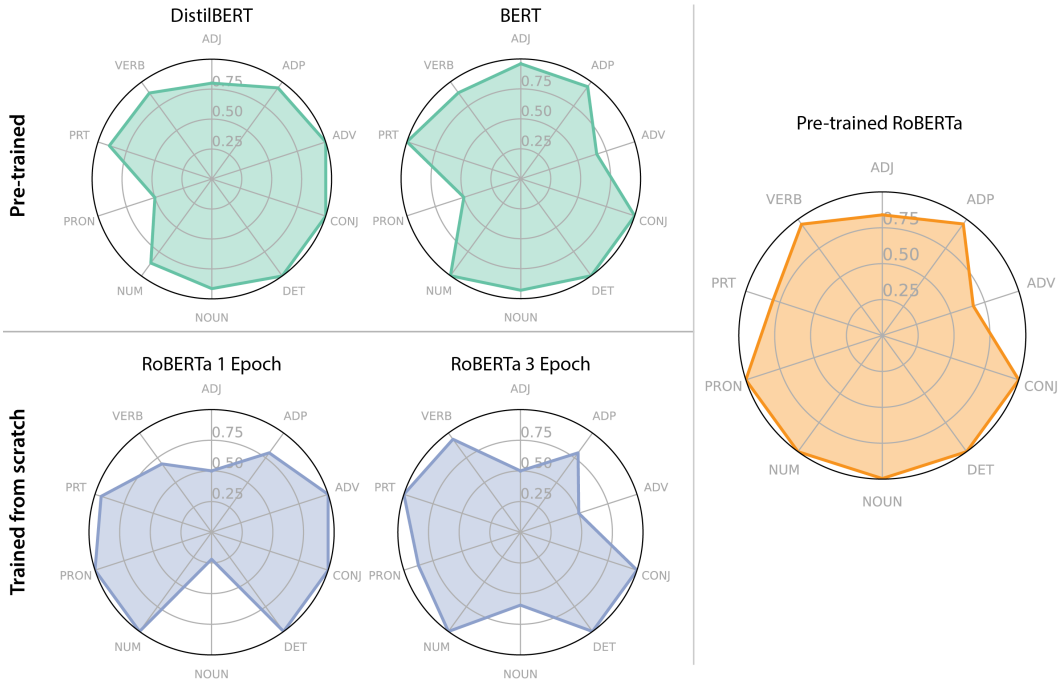

Figure 2: Micro-accuracy of POS classes for pretrained DistilBERT and BERT (green) and for the first two training phases of RoBERTa model (blue), in comparison with pretrained RoBERTa (orange) on the SQuAD dataset.

Model	ADJ	ADP	ADV	CONJ	DET	NOUN	NUM	PRON	PRT	VERB
BERT	-4	5	**-33.3**	0	0	6.9	0	-50	0	**11.2**
DistilBERT	**20**	6	0	0	0	8.5	**-13.3**	50	10	11.2
RoBERTa 1e	-49.9	**19.9**	0	0	0	**-77.9**	0	0	-5.2	-31.1
RoBERTa 3e	**-49.9**	20	**49.9**	0	0	-40.5	0	12.5	0	-6.3
RoBERTa	**16**	4	**-33.3**	0	0	-0.3	0	0	-20	4.2

Table 4: POS tagging experiment results (percent difference from ground truth for a specific POS tag category) on pretrained models (BERT, DistilBERT, and RoBERTa) and trained RoBERTa (1 epoch, 3 epochs) for SQuAD.

4.2.1 How can we quantitatively compare language model knowledge acquisition?

In Table 2 we show graph comparison scores for 3 pretrained BERT variants and 4 trained RoBERTa models across the Google-RE Place of Birth dataset. As discussed in the previous sections, we compute graph similarity between all the models and pretrained RoBERTa using two metrics: an approximation of the graph-edit-distance (Assignment Edit Distance) between extracted graphs and the euclidean distance between the embeddings of the graphs [34, 29]. Notice that BERT and DistilBERT are consistently closer to RoBERTa compared to the trained-from-scratch models (as evidenced by smaller graph-edit-distance and euclidean distance scores). Also note that the both the graph-edit-distance metric (141.25, 135, 130.50, 121.50) and the Euclidean distance of the Graph2Vec representations (0.2260, 0.1733, 0.1607, 0.1605) decrease as the epochs of trained-from-scratch RoBERTa increases. Additional manually calculated graph-edit-distance results are showcased in Appendix Tables 5 and 6. This analysis is important because it enables us to quantitatively compare the knowledge a language model has across model variations and training stages.

4.2.2 As a language model trains, what linguistic traits does it learn over time?

In Table 4 and Appendix Table 7, we present the results of our Part of Speech (POS)-tag metrics, comparing each of the language model KGs to the ground truth KGs and identifying which parts-of-speech are being overpredicted and underpredicted. If a language model KG was performing

perfectly, all of the $POSOR$ percentage differences in Table 4 would be 0. Table 3 sets the scene for the LAMA SQuAD dataset, extracting the parts-of-speech for the ground truth knowledge graph to contextualize the results. *Nouns* are most present with 50.7% of the SQuAD data and *pronouns* are least present with 0.2% of the data. In the case of RoBERTa 1 epoch on the SQuAD dataset, we see that it is severely underpredicting nouns (-77.9%) and overpredicting prepositions (19.9%). In RoBERTa 3 epochs on the SQuAD dataset, we see that we are no longer as drastically underpredicting nouns (-40.6%) and are overpredicting adverbs more than prepositions (49.9%).

Observing the relationships between the pretrained DistilBERT, BERT, and RoBERTa on the Google-RE datasets in Tables 4 and 7, we see that BERT and DistilBERT are very similar (noting that DistilBERT is a distilled form of our BERT-base-cased model). In the Google-RE Place of Birth dataset, the difference in BERT and DistilBERT's largest weaknesses (predicting numbers) and strengths (predicting particles) are less than half of a percent (-25.8% vs. -25.7%, 16.8% vs. 16.3%). In the Google-RE Place of Death dataset, we see that pretrained RoBERTa is much more accurate at predicting pronouns than BERT (0% vs. -33.3%). POS annotations extracted from the KG creation pipeline allow us linguistic granularity in comparing models across various epochs and variants. It also enables machine learning practitioners to produce specialized subsets of data to fix language model linguistic weaknesses in future training iterations.

Using the generated knowledge graphs, we can examine how certain relations change over time. In the RoBERTa trained for 1 epoch, a new relation states {*teachers, follow, as curricula*}. In the 3 epoch version, that same triple changes to {*teachers, follow, curricula*}, which demonstrates a qualitative example of a better grasp of adverbs. In the 1 epoch RoBERTa on the SQuAD dataset, 23 of 40 predicted target objects in the new LM relations were articles (e.g. in, on, the). In 3 epoch RoBERTa, we see a marked reduction in article prediction, with only 9 of 40 predicted target objects as articles.

Lastly, the radio graphs in Figure 2 demonstrate the POS results of RoBERTa at 1 and 3 epochs, as well as pretrained DistilBERT and BERT in comparison with pretrained RoBERTa. As the overall area of the plots increase, we note that the accuracy of the model performance in the corresponding part-of-speech categories increases as well. These analyses are conducted as a proof-of-concept on early stages of RoBERTa and pretrained BERT variants, but the metrics introduced here are broadly applicable and useful for comparing any set of language models.

5 Conclusion

In this paper, we present a pipeline to extract Knowledge Graphs (KGs) from masked language models using cloze statements, and 3 metrics to analyze these graphs comparatively. Our main contribution is the novelty of comparing KG extracts over time to showcase comparative insights between BERT variants at different stages of training. We are inspired by prior work from Petroni et al. [12] to develop our extraction pipeline and expand further on their work through the lens of interpretability and a simpler relation extraction methodology. Our results demonstrate that in each training phase, language models tend to create better connections and more meaningful triples across their extracted knowledge graph, which justifies our intuition towards the performance increase across our metrics in each training epoch (RoBERTa 1 through 7 epochs) and pretrained model advancement (DistilBERT, BERT, RoBERTa). These generated knowledge graphs are a large step towards addressing the research questions: How well does my language model perform in comparison to another (using metrics other than accuracy)? What are the linguistic strengths of my language model? What kind of data should I train my model on to improve it further? Our pipeline aims to become a diagnostic benchmark for language models, providing an alternate approach for AI practitioners to identify language model strengths and weaknesses during the model training process itself.

5.1 Future Work

This work can most certainly be extended further into the realm of other language models and other domains. If we had more resources, a natural extension would be to train RoBERTa from scratch at much larger epochs (i.e. 50, 500, 5000) to better understand later stages of the model training process, instead of looking at only the first 7 epochs. We skimmed the surface of probing task literature to identify skills present in the query answering; delving further into Kim et al. [30]'s benchmarks and

expanding on our POS tagging work could provide more interesting insights. Additionally, switching our model task from masking to sentence generation and feeding it into the same KG extraction pipeline could reveal alternative conclusions about grammar and linguistic strengths. Improving the dataset size and variation is another important step; more diverse data extracted from Wikipedia could provide stronger evaluation-based claims [40].

5.2 Journal Submission

The authors would like an extended version of this submission to be considered for publication in a journal special issue.

References

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[2] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018.

[3] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016.

[4] Adina Williams, Nikita Nangia, and Samuel R Bowman. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*, 2017.

[5] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

[6] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[7] Sebastian Ruder. Ml and nlp research highlights of 2020. `https://ruder.io/research-highlights-2020/index.html#5-evaluation-beyond-accuracy`, 2020. (Accessed on 02/08/2021).

[8] Jinlan Fu, Pengfei Liu, and Graham Neubig. Interpretable multi-dataset evaluation for named entity recognition. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6058–6069, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.489. URL `https://www.aclweb.org/anthology/2020.emnlp-main.489`.

[9] Divyansh Kaushik, Eduard Hovy, and Zachary Lipton. Learning the difference that makes a difference with counterfactually-augmented data. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=Sklgs0NFvr`.

[10] Matt Gardner, Yoav Artzi, Victoria Basmova, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, Nitish Gupta, Hanna Hajishirzi, Gabriel Ilharco, Daniel Khashabi, Kevin Lin, Jiangming Liu, Nelson F. Liu, Phoebe Mulcaire, Qiang Ning, Sameer Singh, Noah A. Smith, Sanjay Subramanian, Reut Tsarfaty, Eric Wallace, Ally Zhang, and Ben Zhou. Evaluating models' local decision boundaries via contrast sets, 2020.

[11] Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. Blimp: The benchmark of linguistic minimal pairs for english. *Transactions of the Association for Computational Linguistics*, 8:377–392, 2020. doi: 10.1162/tacl_a_00321. URL `https://doi.org/10.1162/tacl_a_00321`.

[12] Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H Miller, and Sebastian Riedel. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2019*, 2019.

[13] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi 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 M. Rush. 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, October 2020. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/2020.emnlp-demos.6`.

[14] John Hewitt and Christopher D. Manning. A structural probe for finding syntax in word representations. 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 4129–4138, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1419. URL `https://www.aclweb.org/anthology/N19-1419`.

[15] Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. What does bert look at? an analysis of bert's attention, 2019.

[16] Yoav Goldberg. Assessing bert's syntactic abilities, 2019.

[17] Andy Coenen, Emily Reif, Ann Yuan, Been Kim, Adam Pearce, Fernanda Viégas, and Martin Wattenberg. Visualizing and measuring the geometry of bert. *arXiv preprint arXiv:1906.02715*, 2019.

[18] Ian Tenney, Dipanjan Das, and Ellie Pavlick. Bert rediscovers the classical nlp pipeline, 2019.

[19] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1202. URL `https://www.aclweb.org/anthology/N18-1202`.

[20] Yaru Hao, Li Dong, Furu Wei, and Ke Xu. Visualizing and understanding the effectiveness of bert, 2019.

[21] Leo Z. Liu, Yizhong Wang, Jungo Kasai, Hannaneh Hajishirzi, and Noah A. Smith. Probing across time: What does roberta know and when?, 2021.

[22] Cheng-Han Chiang, Sung-Feng Huang, and Hung yi Lee. Pretrained language model embryology: The birth of albert, 2020.

[23] Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.442. URL `https://www.aclweb.org/anthology/2020.acl-main.442`.

[24] Fabio Petroni, Patrick Lewis, Aleksandra Piktus, Tim Rocktäschel, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. How context affects language models' factual predictions. In *Automated Knowledge Base Construction*, 2020. URL `https://openreview.net/forum?id=025X0zPfn`.

[25] Benjamin Heinzerling and Kentaro Inui. Language models as knowledge bases: On entity representations, storage capacity, and paraphrased queries, 2021.

[26] Carlos Aspillaga, Marcelo Mendoza, and Alvaro Soto. Tracking the progress of language models by extracting their underlying knowledge graphs, 2021. URL https://openreview.net/forum?id=ghKbryXRRAB.

[27] Ankur Goswami, Akshata Bhat, Hadar Ohana, and Theodoros Rekatsinas. Unsupervised relation extraction from language models using constrained cloze completion, 2020.

[28] Alberto Sanfeliu and King-Sun Fu. A distance measure between attributed relational graphs for pattern recognition. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-13(3): 353–362, 1983. doi: 10.1109/TSMC.1983.6313167.

[29] Annamalai Narayanan, Mahinthan Chandramohan, Rajasekar Venkatesan, Lihui Chen, Yang Liu, and Shantanu Jaiswal. graph2vec: Learning distributed representations of graphs. *arXiv preprint arXiv:1707.05005*, 2017.

[30] Najoung Kim, Roma Patel, Adam Poliak, Patrick Xia, Alex Wang, Tom McCoy, Ian Tenney, Alexis Ross, Tal Linzen, Benjamin Van Durme, Samuel R. Bowman, and Ellie Pavlick. Probing what different NLP tasks teach machines about function word comprehension. In *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (*SEM 2019)*, pages 235–249, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/S19-1026.

[31] Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. spaCy: Industrial-strength Natural Language Processing in Python, 2020. URL https://doi.org/10.5281/zenodo.1212303.

[32] Textacy library. https://pypi.org/project/textacy/. (Accessed on 02/10/2021).

[33] 50,000 lessons on how to read: a relation extraction corpus. https://ai.googleblog.com/2013/04/50000-lessons-on-how-to-read-relation.html. (Accessed on 02/08/2021).

[34] Kaspar Riesen and Horst Bunke. Approximate graph edit distance computation by means of bipartite graph matching. *Image and Vision computing*, 27(7):950–959, 2009.

[35] Benedek Rozemberczki and Rik Sarkar. Characteristic functions on graphs: Birds of a feather, from statistical descriptors to parametric models. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 1325–1334, 2020.

[36] Naomi Saphra and Adam Lopez. Language models learn pos first. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 328–330, 2018.

[37] Christopher D Manning. Part-of-speech tagging from 97% to 100%: is it time for some linguistics? In *International conference on intelligent text processing and computational linguistics*, pages 171–189. Springer, 2011.

[38] Atro Voutilainen. Part-of-speech tagging. *The Oxford handbook of computational linguistics*, pages 219–232, 2003.

[39] Edward Loper and Steven Bird. Nltk: The natural language toolkit. In *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics - Volume 1*, ETMTNLP '02, page 63–70, USA, 2002. Association for Computational Linguistics. doi: 10.3115/1118108.1118117. URL https://doi.org/10.3115/1118108.1118117.

[40] Luyu Wang, Yujia Li, Ozlem Aslan, and Oriol Vinyals. Wikigraphs: A wikipedia text - knowledge graph paired dataset, 2021.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] **See Section 5.1.**
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes] **As we are using publicly released datasets, cite related work, and use pre-published models, we can confirm no additional ethical concerns generated from this work.**
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] **All of our experiments are available at https://anonymous.4open.science/r/interpret-lm-2021/, and will be released publicly on Github upon acceptance.**
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] **See Section 3.1.**
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A] **Our experiments do not have accuracy scores or traditional metrics.**
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] **See Section 3.1.**
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] **We build off of existing literature (LAMA, graph2vec, BERT variations) and cite each paper where appropriate.**
   (b) Did you mention the license of the assets? [Yes] **It is mentioned in the repository; the datasets and models are not new to this work.**
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] **All of our code, data, and parsers are available in our repository.**
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

Appendix

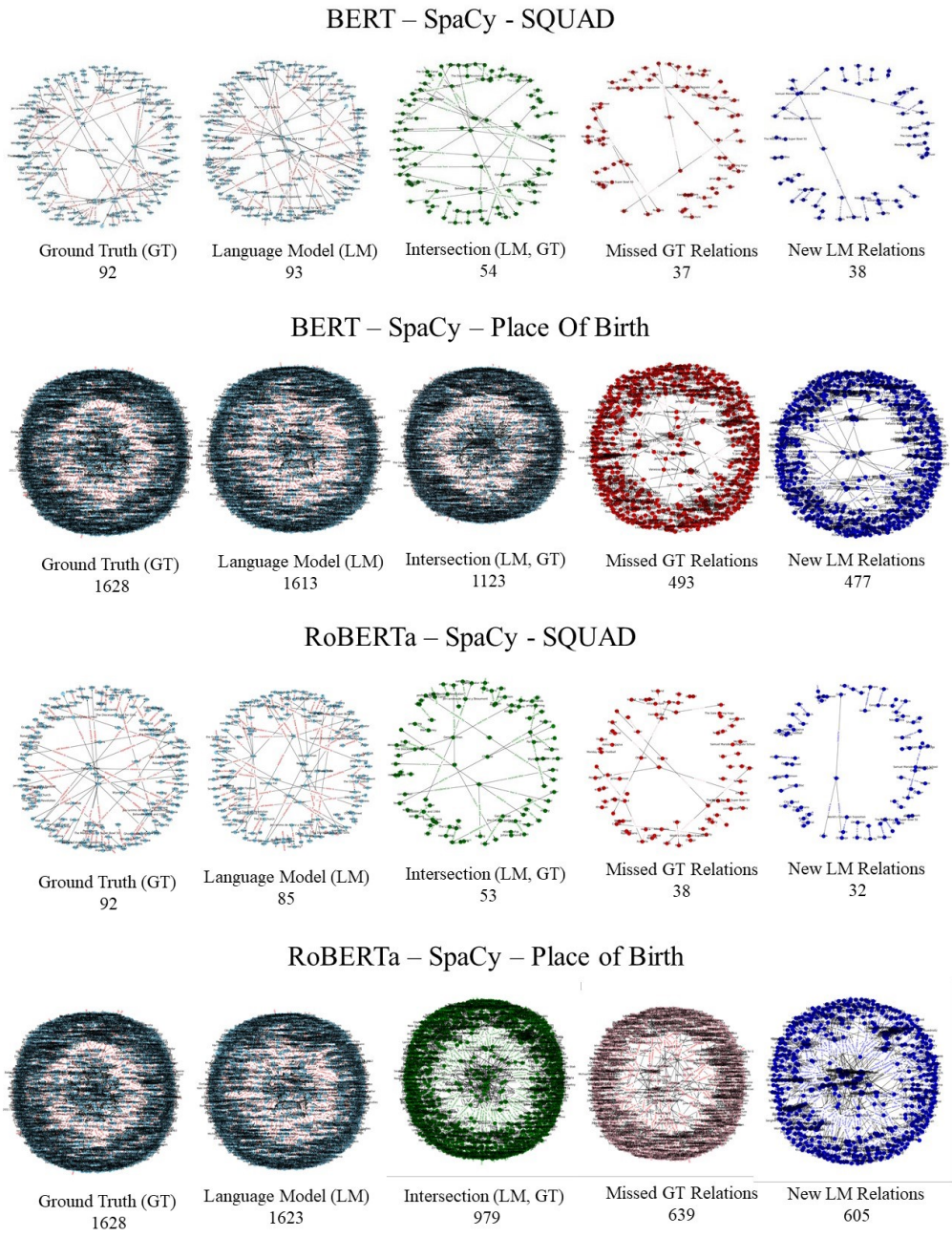

Figure 3: Knowledge Graphs (KGs) generated using the pretrained BERT / RoBERTa architecture and SpaCy relation extraction on the LAMA SQUAD and Google RE datasets. The numbers below each KG represent the number of unique relation triples. The LM KG represents the cloze sentences after the masks have been replaced by the Language Model predictions. Ground Truth references gold label sentences where relations have been extracted directly.

SQuAD	MobileBERT	DistilBERT	BERT	RoBERTa
MobileBERT	0	15	21	25
DistilBERT	15	0	12	23
BERT	21	12	0	17
RoBERTa	25	23	17	0

Table 5: Graph-Edit-Distance scores between KGs for BERT variants using the SQUAD dataset from LAMA [12].

Google-RE	DistilBERT	BERT	RoBERTa
DistilBERT	0	113	272
BERT	113	0	287
RoBERTa	272	287	0

Table 6: Graph-Edit-Distance scores between KGs for BERT variants using the Google-RE Place of Birth Dataset from LAMA [12].

POS-tag Experiment	.	ADJ	ADP	ADV	CONJ	DET
re-date-birth BERT	**-5.3**	-1.1	0.6	0	0	6.2
re-date-birth RoBERTa 1e	0	-6.2	0	0	0	**66.6**
re-date-birth RoBERTa 3e	0	-6.2	**-33.3**	0	0	**53.8**
re-date-birth RoBERTa (pretrain)	**52.6**	-1.7	0.9	5	9	2
re-place-birth BERT	0.5	-0.4	-2.1	3.1	0	2.5
re-place-birth DistilBERT	1.5	-1.7	-2.1	3.1	0	1.6
re-place-birth RoBERTa 1e	0	-4	**-37.4**	0	0	0
re-place-birth RoBERTa 3e	0	-4	**-37.4**	0	0	0
re-place-birth RoBERTa (pretrain)	-2.5	-8.9	3.6	-3.1	3.4	1.6
re-place-death BERT	20.6	19.2	**27.1**	-19	14.2	20.5
re-place-death DistilBERT	22.2	19.2	**27.9**	**-14.2**	14.2	20.5
re-place-death RoBERTa (pretrain)	20.6	15.3	**33**	**-19**	14.2	17.6
squad BERT	0	-4	5	**-33.3**	0	0
squad DistilBERT	0	**20**	6	0	0	0
squad RoBERTa 1e	0	-49.9	**19.9**	0	0	0
squad RoBERTa 3e	0	**-49.9**	19.9	**49.9**	0	0
squad RoBERTa (pretrain)	0	**16**	4	**-33.3**	0	0

POS-tag Experiment	NOUN	NUM	PRON	PRT	VERB
re-date-birth BERT	-1.9	**14.4**	0	-1.8	1.9
re-date-birth DistilBERT	-0.6	8	0	1.8	1.2
re-date-birth RoBERTa 1e	12.4	**-74.9**	-0.8	0	-3.6
re-date-birth RoBERTa 3e	5.9	0	0	0	-9.2
re-date-birth RoBERTa (pretrain)	**-7.6**	60	0	-1.2	9.2
re-place-birth BERT	-2.2	**-25.7**	14.2	**16.8**	-1.7
re-place-birth DistilBERT	-1.9	**-25.7**	14.2	**16.3**	-1.9
re-place-birth RoBERTa 1e	-0.5	0	0	**5.5**	0.7
re-place-birth RoBERTa 3e	-1.2	0	0	**5.5**	-0.3
re-place-birth RoBERTa (pretrain)	3.9	**-24.3**	14.2	**17.3**	3.5
re-place-death BERT	24.2	12.2	**-33.3**	17.4	26.5
re-place-death DistilBERT	25.2	13.6	0	17.4	26.8
re-place-death RoBERTa (pretrain)	28.8	19.3	0	19	31.5
squad BERT	6.9	0	-50	0	**11.2**
squad DistilBERT	8.5	**-13.3**	50	10	11.2
squad RoBERTa 1e	**-77.9**	0	0	-5.2	-31.1
squad RoBERTa 3e	-40.5	0	12.4	0	-6.3
squad RoBERTa (pretrain)	-0.3	0	0	-20	4.2

Table 7: Difference in POS tags between the language-model-generated KG and ground truth KG. The largest positive and negative difference are highlighted. POS-tags use the NLTK Averaged Perceptron Tagger abbreviations: VERB - verbs (all tenses and modes), NOUN - nouns (common and proper), PRON - pronouns, ADJ - adjectives, ADV - adverbs, ADP - adpositions (prepositions and postpositions), CONJ - conjunctions, DET - determiners, NUM - cardinal numbers, PRT - particles or other function words, . - punctuation [39].