Interpretability of BERT Latent Space through Knowledge Graphs

Vito Walter Anelli
Politecnico di Bari
Bari, Italy
vitowalter.anelli@poliba.it

Giovanni Maria Biancofiore*
Politecnico di Bari
Bari, Italy
giovannimaria.biancofiore@poliba.it

Alessandro De Bellis*
Politecnico di Bari
Bari, Italy
a.debellis6@studenti.poliba.it

Tommaso Di Noia
Politecnico di Bari
Bari, Italy
tommaso.dinoia@poliba.it

Eugenio Di Sciascio
Politecnico di Bari
Bari, Italy
eugenio.disciascio@poliba.it

ABSTRACT

The advent of pretrained language have renovated the ways of handling natural languages, improving the quality of systems that rely on them. BERT played a crucial role in revolutionizing the Natural Language Processing (NLP) area. However, the deep learning framework it implements lacks interpretability. Thus, recent research efforts aimed to explain what BERT learns from the text sources exploited to pre-train its linguistic model. In this paper, we analyze the latent vector space resulting from the BERT context-aware word embeddings. We focus on assessing whether regions of the BERT vector space hold an explicit meaning attributable to a Knowledge Graph (KG). First, we prove the existence of explicitly meaningful areas through the Link Prediction (LP) task. Then, we demonstrate these regions being linked to explicit ontology concepts of a KG by learning classification patterns. To the best of our knowledge, this is the first attempt at interpreting the BERT learned linguistic knowledge through a KG relying on its pretrained context-aware word embeddings.

CCS CONCEPTS

• **Information systems** → **Information retrieval**; **Ontologies**; **Document topic models**.

KEYWORDS

Natural Language Processing, Deep Learning, Knowledge Graphs

ACM Reference Format:
Vito Walter Anelli, Giovanni Maria Biancofiore, Alessandro De Bellis, Tommaso Di Noia, and Eugenio Di Sciascio. 2022. Interpretability of BERT Latent Space through Knowledge Graphs. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management (CIKM '22), October 17–21, 2022, Atlanta, GA, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3511808.3557617

*Authors are listed in alphabetical order. Corresponding authors: Giovanni Maria Biancofiore, Alessandro De Bellis.

CIKM '22, October 17–21, 2022, Atlanta, GA, USA

ACM ISBN 978-1-4503-9236-5/22/10…$15.00
https://doi.org/10.1145/3511808.3557617

1 INTRODUCTION

Natural Language Processing (NLP) has experienced several radical changes in its paradigms. Large amounts of linguistic data have promoted deep learning models to learn textual data representations at the expense of hand-crafted feature engineering approaches. This led to the design of highly successful architectures in implementing language models (i.e., Benjo et al. [1] and Mikolow et al. [12]). The growing interest of the research community, in parallel with the development of deep learning, has contributed to the explosion of a massive variety of NLP models to accomplish the most diverse applications [8]. The work of Vaswani et al. [25] stood out for its performance in solving the sequence transduction task with Transformers. Delvin et al. [6] got inspired by both this architecture and the gain in popularity of the pre-training and fine-tuning formula recorded in those years [13]. They proposed Bidirectional Encoder Representations from Transformers (BERT), which draw a turning point in state-of-the-art in NLP. Unlike the earlier proposed pre-trained language models [14, 9, 16], BERT implements a masked paradigm based on the Cloze task [21] and a next sentence prediction assignment to pre-train its language model. Such configuration, once fine-tuned, allowed BERT to achieve competitive performances on multiple benchmarks (e.g., GLUE [27] and SQuAD [18, 17]), which triggered the generation of many BERT variants to improve the resolution of the most popular NLP-based tasks.

Xia et al. [33] collected all the BERT implementations in five areas of progress. Four of them own most published works on modifying BERT to reach specific goals, i.e., improving the language model acting on the pre-training objectives or data, the model efficiency, and the multilingual ability. In contrast, the fifth field on interpretability hold fewer works, reflecting the non-trivial task of interpreting such a sophisticated framework. Nevertheless, recent trends in published papers show an increasing interest in this topic. Many works inspect the model through attention heads [26, 4, 20, 10, 30], fine-tune it to solve interpretable tasks [11], or even modify the pre-training procedure for the same goals [22, 32].

On the other hand, only a minority poses questions on the semantic space conformations resulting from BERT. Some of them rely on classifiers to assess held information about the Entity-Linking [3], entity category clustering [5], and link prediction [15] tasks, which provide insights into the BERT learned knowledge. However, these solutions require authors to modify the embedding representations or target a multiclassification job. Thus, they are more proper to consider which information BERT learns to distinguish the embeddings rather than providing data on the properties of

their space. Conversely, Ethayarajh [7] first studies the BERT latent space properties by comparing the cosine similarity between contextualized pre-trained BERT word embeddings. These word representations result from feeding BERT with words enclosed in contextual sentences without fine-tuning it. From this point, we will refer to this type of embeddings simply as BERT embeddings. Ethayarajh states that BERT vector representations are anisotropic concerning their direction, forming groups in narrow cones.
In this work, we aim to answer the following research questions:

- R1: Does BERT generate a latent semantic space holding information about knowledge with explicit semantics?
- R2: Can we learn functions to automatically detect precise knowledge graph concepts from the BERT latent space?

The study aims to investigate the BERT embeddings space looking for the existence of meaningful regions about explicit concepts and their edges. To this purpose, we rely on knowledge graphs (KGs) that connect entities via directed edges with an explicit semantics (relations). The resulting semantic network can be serialized via a set of triples subject-predicate-object where subject and objects represent unambiguous entities with a unique identifier. Given a KG, we first compare the behaviors of the BERT embeddings with several types of KG embeddings through Link Prediction (LP). Our intuition is that similarities between the two types of embeddings highlight that the BERT space contains the inherent structure of the KG. Also, they reflect the BERT embeddings holding explicit semantic information that depends on their position, thus areas that affect such information (i.e., a topology). Then, we prove this property leads to exact KG concepts (i.e., ontological classes) by learning patterns on BERT embeddings through several classifiers, one for each KG ontological category. Differently from a single classifier trained on a multi-classification task, individual binary classifiers extract feature patterns that uniquely identify each concept instead of checking BERT embedding differences for the assignment task. In other words, binary classifiers model regions in the BERT space that refer to the ontological categories. In contrast, the multi-classifier learns which features distinguish the word representations among the finite set of concepts, discarding the existence of other categories or the simultaneous belonging to multiple categories.

As the main contribution, we prove that BERT embeddings already possess information about the structure of a KG without needing any fine-tuning processes or architectural changes. We demonstrate that inner spatial properties of the BERT vector space allow inferring explicit KG concepts. Extensive experiments support our findings, which are available here[1] for reproducibility.

2 METHODOLOGY

The first goal we address is to prove the existence of explicit semantics behind the latent space resulting from BERT embeddings. BERT representations have been shown to encode syntactic and hierarchical information (e.g., parts of speech, syntactic functions, subject-predicate agreements). Also, they are aware of semantic roles, entity types, and relations derived from linguistic knowledge [19]. However, the BERT vector space is mostly unexplored. From [7], we learn that the BERT space is anisotropic. Instead, Dalvi et al. [5] show that BERT embeddings can be grouped according to interpretable, but not explicit concepts. Thus, recent literature does not let us assert that the BERT space contains explicit semantics.

Our approach analyzes similarities of the BERT embeddings with standard KG embeddings in solving the LP task. In detail, KG embeddings encode explicit semantics of a KG into a continuous vector space preserving its inherent structure [28]. At the same time, the LP focuses on predicting the correctness of unseen triplets (i.e., subject-predicate-object), which also identifies a way to test the learning of a KG structure [24]. Thus, testing BERT embeddings with the LP will give further insights into the clear semantic information and the KG structure they may hold. To make our study significant, we lead our analysis under the following assumptions:

(1) The BERT embeddings result from the BERT vanilla pre-trained model;
(2) KG embeddings come out of training where both the subject and the object entities, of a subject-predicate-object triple in a KG, share the same vector space;
(3) All the embeddings possess the same dimension.

The assumption (1) allows us to evaluate the intrinsic capability of BERT to encode explicit semantic information in its pre-trained language model, thus making deductions on its derived latent space. Fine-tuning procedures may specialize information held by BERT, which affects the generality of our study. Therefore, we do not use variants like KG-BERT [35] or the one proposed by Petroni et al. [15] since they modify the pre-trained language model or fine-tune BERT. The last two assumptions make the comparisons between the BERT embeddings and the KG embeddings significant. On the one hand, with assumption (2), we are interested in investigating only the vector space resulting from BERT; thus, KG embeddings and LP methods that require a space transformation or additional support spaces are out of our scope. On the other hand, the constraint (3) places the examined embeddings on equal terms.

Given a KG, we compute the BERT embeddings on its entities. Since the most effective BERT representations in a semantic task are context-aware, we feed the BERT model with the entity label plus a related context. In particular, we compose sentences in the form of "label + *be* + abstract" for each KG entity. Then, we gain the contextualized BERT representation from the last hidden units that refers to the label. It is worth noting that we do not make any assumptions about the granularity of the entity labels. Indeed, BERT operates on the WordPiece [31] segmentation of the input on a sub-word level, mainly to avoid the mismatch vocabulary issue. Hence, each BERT hidden units represent single sub-words. The BERT embedding of the label results from aggregating its sub-word hidden units, which can belong to one or more words.

More precisely, let w be a word made of a set of WordPiece tokens t s.t. $w = [t_1, t_2, ..., t_n]$ and l be the entity label, which may have one o more words s.t. $l = [w_1, w_2, ...w_m]$. Thus, we refer with t_{ji} to the i-th WordPiece token of the j-th word, with $1 \leq i \leq n$ and $1 \leq j \leq m$. Given the BERT hidden unit h, the BERT embedding of a WordPiece token t results form $h_s(t)$, including the contextual information derived from the whole input sentence s. Therefore, $h_s(t_{ji})$ identifies the BERT embedding for the i-th WrodPiece token of the j-th word. In our approach, we denote with b the single word *be* and with c the set of d words composing the entity abstract. Thus, we can refer to the input that we give to BERT for encoding each

[1] https://bit.ly/3T0987I

entity with $s = l + b + c$. Referring with w^l and t^l respectively to words and WordPiece tokens that belong to the entity label l, given an aggregating functions f, the BERT embedding of an entity $\mathbf{e}$ derives from the aggregation of the WordPiece tokens embeddings of each word belonging to its label s.t. $\mathbf{e} = f(h_s(t^l_{ji}))$.

Once we encoded all the KG entities, we learn the relation embeddings over their entity BERT embeddings through existing LP models. We have chosen not to use BERT embeddings for relations as they differ from entities as semantic components. In an anisotropic space, such representations would depend both on the entities and relations directions. In this way, we avoid assuming that relationships and entities have similar properties or they are analogous elements. To accomplish requirement (2), we opt for three well-established LP models such as TransE [2], TransH [29], and DistMult [34]. TransE is a translational distance model defining both entities and relations as vectors in the same space. Given a KG fact (h, r, t), the relation is interpreted as a translation vector $\mathbf{r}$ so that the embedded entities $\mathbf{h}$ and $\mathbf{t}$ can be connected with $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$. Similarly, TransH follows the intuition of TransE by introducing relation-specific hyperplanes defined by the normal vector $\mathbf{w}_r$. Therefore, we have to first project the entity representations $\mathbf{h}$ and $\mathbf{t}$ onto the hyperplane to have their connection: $\mathbf{h}_\perp = \mathbf{h} - \mathbf{w}_r^T\mathbf{h}\mathbf{w}_r$, $\mathbf{t}_\perp = \mathbf{t} - \mathbf{w}_r^T\mathbf{t}\mathbf{w}_r$. Instead, DistMult represents each relation as a diagonal matrix $\mathbf{M}_r = \text{diag}(\mathbf{r})$ modeling pairwise interactions between components of $\mathbf{h}$ and $\mathbf{t}$ along the same dimensions with a scoring function: $f_r(h, t) = \mathbf{h}^T\text{diag}(\mathbf{r})\mathbf{t}$. We apply these models directly to the BERT embeddings of the KG entities to infer their relations according to the KG structure. The BERT embeddings used for the entities remain fixed during the whole training process. Since the LP models encode the KG explicit semantics in the resulting KG embeddings, similarities of performances on the LP task for the KG embeddings and the BERT ones gives an answer to our research question R1 in Section 1.

The second objective is to assess whether there exist properties of the BERT space that enable the detection of precise KG concepts. To find those BERT space properties, we train several binary classifiers to detect if the BERT embedding of an entity belong to an ontological class. We design each classifier to recognize a single concept, implementing one classifier for each KG class. In detail, this type of classifier will learn patterns from the BERT representations related to the concerned ontological concept. In fact, these classifiers know no other class than their own. Thus, they focus only on the input features (i.e., BERT embeddings) that allow deriving their own class. These features properties can then be extended to the whole embedding space. Reaching high test performances on each classifier answers to research question R2 in Section 1.

3 SEMANTIC ANALYSIS

This section provides the detailed configuration of the experiments that led to our analysis of the BERT space. We explore its inner properties through the embeddings computed by the pre-trained language model of the BERT-Base-cased. We choose to investigate the cased version since we believe that cased words enclose a different semantic than uncased ones.

We use Freebase (FB15k-237) [23] as the benchmark dataset to implement LP over the BERT embeddings. The main intuition here is that BERT already contains Freebase explicit semantics since it was pre-trained on the English Wikipedia corpus where Freebase was built. Since we need to feed BERT with sentences formed concatenating the label, the verb *be*, and the abstract for each entity, we discard from the FB15k-237 all the entities with no label or abstract in their Wikidata mapping[2]. Thus, we obtain an FB15K-237 subset by removing the facts related to the discarded entities, and we call it FB15K-237-Desc. It contains 266,263 facts over 13,667 entities compared to FB15K-237, which has 310,116 facts over 14,541. For completeness, we also compute separated entities BERT representations by feeding the BERT model with only the entities labels. These embeddings will benchmark the utility of the context for BERT in positioning them into the most appropriate space region. In both cases, the aggregation of the hidden units of the WordPiece tokens referring to the entities takes place through the arithmetic mean function.

Link Prediction and Semantics. Once we have computed the two BERT representations for all the FB15K-237-Desc entities, respectively BL (i.e., BERT Label) and BD (i.e., BERT Desc), we start the LP task in three configurations. The first one computes the standard KG embedding of TransE, TransH, and DistMult to compose our baselines. Their performance needed to be recalculated to accomplish the requirement (3) since their embeddings size is 768. The second setting learns the relation embeddings over the BL entities representations, which enable the evaluation of the LP over the BL entity embeddings. The last scenario differs in the computation of the BERT entity embeddings through contextualized sentences (BD). All the configurations have their models trained on FB15K-237-Desc, which is split into 80%-10%-10% to generate the train, evaluation, and test sets. The training procedure follows the minibatch mode over the raw and filtered negative sampling proposed by Bordes et al. [2].

	Raw			Filtered		
	MR	hit@10	hit@5	MR	hit@10	hit@5
TransE	**305.32**	**33.15**	**24.48**	**177.86**	**45.94**	**37.19**
TransH	412.73	29.74	21.39	271.88	40.98	32.39
DistMult	395.48	25.46	17.30	281.00	33.79	24.84
TransE$_{BL}$	866.47	21.30	15.88	773.43	25.03	19.56
TransH$_{BL}$	968.67	20.98	15.88	875.44	24.54	19.62
DistMult$_{BL}$	847.86*	19.84*	14.61*	753.46*	23.76*	18.09*
TransE$_{BD}$	604.83	23.26	17.04	508.40	28.68	22.15
TransH$_{BD}$	702.62	20.62	15.49	609.64	24.84	19.01
DistMult$_{BD}$	560.64*	21.31*	15.60*	467.01*	26.11*	19.83*

Table 1: Evaluation results of the LP task through the standard, BERT label (BL) and BERT description (BD) embeddings of TransE, TransH and DistMult. In bold, the best achieved results over the three configurations. The asterisks mark the BL and BD results closest to those of the standard model. The underlined outcomes highlight those values most relative to the best results.

We used the grid-search with early stopping to select the hyperparameters leading to the best performance in each configuration. The batch size has a value fixed to 1200 while the margin value γ can assume values among (1, 2, 10) for TransE, (0.25, 0.5, 1, 2) for TransH and (0.001, 0.005, 0.01) for DistMult. In addition, TransH has its soft constraint weight selected among (0.015625, 0.0625, 0.25, 1), and TransE has its distance function tested between L1 and L2. Both TransE and TransH were trained with stochastic gradient

[2]https://developers.google.com/freebase/

descent (SGD) in the first LP configuration, while in the remaining two settings they exploit the Adam optimizer. DitsMult instead uses the AdaGrad optimizer in each LP scenario. We adopt the Mean Rank (MR) and the hit@n (i.e., hit@10 and hit@5) metrics to assess the LP performances. Table 1 resumes the results we achieved.

Surprisingly, the standard TransE reaches the best results in each configuration. This outcome derives from its known ability to catch the KG geometrical properties. In contrast, TransH and DistMult collect more semantic information, which can behave like noises given the different embedding dimensionality. The BL embeddings instead obtain the worst performances as we expected, further proving the importance of the context in correctly positioning the BERT representations in its vector space. The last configuration gives the most exciting results. Albeit maintaining the same behaviors of the models, TransE over the BD embeddings gets comparable results with the standard DistMult model. This finding shows that we can infer meaningful explicit semantics by referring to the BERT embeddings' relevant properties (i.e., specific space dimensions). Therefore, we can say that the BERT space intrinsically contains a KG structure and precise semantics.

Ontological Analysis. The second experiment trains binary classifiers for each KG ontological class to detect whether a contextualized BERT embedding belongs to a specific category. Thus, we first retrieve from the Wikidata mapping all the entities' ontological classes through their *instanceOf* property, discarding those having no label or category. We use the FB15K [2] since it contains more entities than its smaller counterpart, and we limit our observations to those classes with enough entities to train meaningful classifiers. Hence, we obtain 20 categories with 9,258 entities distributed as in Figure 1.

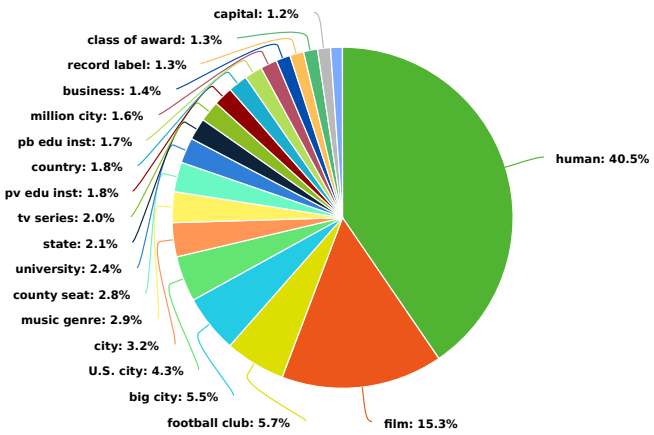

Figure 1: Distribution of the FB15K entities among the Wikidata ontological classes.

It is worth noting how classes like *city*, *big_city* and *US_city* can group into a single category. However, we first split the overall dataset into 80%-10%-10% to generate the train, validation, and test set. In this manner, we avoid classifiers knowing the entity class from the training set during their test. In addition, we perform this partition maintaining the same distribution of classes. Then, for each category, we select from the train set all those samples which belong to the related classifier category, and we added an equal amount of entities from other classes to have the train set balanced. We model each classifier as a feedforward neural network with a single hidden layer of 300 units that uses the ReLU activation function and the Adam optimizer. We evaluate the classifiers' performances through their accuracy, precision, recall, and F1 measure. Table 2 resumes the results of each model and gives data about the positive output for each class through the support.

	Acc. (%)	P (%)	R (%)	F1 (%)	Supp. (#)
human	**99.95**	100	99.88	**99.94**	866
film	99.14	95.33	100	97.61	327
football club	99.78	96.82	100	98.39	122
big city	94.06	50.46	97.32	66.46	112
U.S. city	98.49	76.47	100	86.67	91
city	95.30	43.05	98.48	59.91	66
music genre	99.84	95.38	100	97.64	62
county seat	96.76	50	100	66.67	60
university	95.65	45.04	98.04	61.73	51
state	99.19	75.43	97.73	85.15	44
tv series	93.84	26.62	97.62	41.83	42
pv edu Inst	95.79	33.34	100	50	39
country	99.46	80	100	88.89	40
pb edu Inst	97.35	42.23	100	59.50	36
million city	94.60	23.08	100	37.50	30
business	95.63	26.36	100	41.73	23
record label	97.03	33.73	100	50.45	28
class of award	99.62	80	100	88.89	28
capital	94.55	18.70	95.83	31.29	24
o.a. publisher	95.46	21.70	95.84	35.38	24

Table 2: Evaluation results of the binary classifiers trained on each Wikidata category. In bold, the classifier's best results.

As we can see, all the classifiers reach a high value of accuracy, among which the performance of the *human* classifier emerges. We explain these outcomes since the *human* classifier possesses most of the KG data supporting its modeling. Moreover, the *human* category identifies the most unambiguous class of the dataset, making it easy for its classifier to recognize its entities. Conversely, classes like *city*, *big_city*, and *US_city* have their classifier reaching low precision values despite their high recall. This result mainly depends on how these categories actually identify a single one. Indeed, the high recall highlights that the classifiers recognize all the positive samples of that class. At the same time, the low precision shows that they also identify other entities as belonging to that group. Therefore, We can infer that the BERT embeddings contain precise information that leads to explicit ontological classes, which can be extended to the spatial properties of the BERT vector space.

4 CONCLUSION

We have analyzed the latent vector space resulting from the BERT context-aware word embeddings, assessing whether the BERT vector space regions hold an explicit semantics attributable to Knowledge Graphs (KGs). We have demonstrated the existence of explicitly meaningful areas exploiting the traits of the Link Prediction task. Then, we show these regions lead to explicit ontology concepts of a KG by learning classification patterns over the BERT embeddings. To our knowledge, no previous works attempted to interpret the BERT learned linguistic knowledge through a Knowledge Graph. In future work, we will extend the analysis to other KGs, and unveil the potential common semantic patterns across the different graphs.

Acknowledgements. The authors acknowledge partial support from the projects PASSEPARTOUT, ServiziLocali2.0, Smart Rights Management Platform, BIO-D, and ERP4.0.

REFERENCES

[1] Yoshua Bengio, Réjean Ducharme, and Pascal Vincent. 2000. A neural probabilistic language model. *Advances in Neural Information Processing Systems*, 13.

[2] Antoine Bordes, Nicolas Usunier, Alberto Garcıa-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *NIPS*, 2787–2795.

[3] Samuel Broscheit. 2019. Investigating entity knowledge in BERT with simple neural end-to-end entity linking. In *CoNLL*. Association for Computational Linguistics, 677–685.

[4] Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does BERT look at? an analysis of bert's attention. In *BlackboxNLP@ACL*. Association for Computational Linguistics, 276–286.

[5] Fahim Dalvi, Abdul Rafae Khan, Firoj Alam, Nadir Durrani, Jia Xu, and Hassan Sajjad. 2022. Discovering latent concepts learned in BERT. *CoRR*, abs/2205.07237.

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

[7] Kawin Ethayarajh. 2019. How contextual are contextualized word representations? comparing the geometry of bert, elmo, and GPT-2 embeddings. In *EMNLP/IJCNLP (1)*. Association for Computational Linguistics, 55–65.

[8] Julia Hirschberg and Christopher D Manning. 2015. Advances in natural language processing. *Science*, 349, 6245, 261–266.

[9] Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *ACL (1)*. Association for Computational Linguistics, 328–339.

[10] Sarthak Jain and Byron C. Wallace. 2019. Attention is not explanation. In *NAACL-HLT (1)*. Association for Computational Linguistics, 3543–3556.

[11] Da Li, Sen Yang, Kele Xu, Yukai He Ming Yi, and Huaimin Wang. 2022. Multi-task pre-training language model for semantic network completion. *arXiv preprint arXiv:2201.04843*.

[12] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

[13] Bonan Min, Hayley Ross, Elior Sulem, Amir Pouran Ben Veyseh, Thien Huu Nguyen, Oscar Sainz, Eneko Agirre, Ilana Heinz, and Dan Roth. 2021. Recent advances in natural language processing via large pre-trained language models: a survey. *arXiv preprint arXiv:2111.01243*.

[14] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *NAACL-HLT*. Association for Computational Linguistics, 2227–2237.

[15] Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick S. H. Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander H. Miller. 2019. Language models as knowledge bases? In *EMNLP/IJCNLP (1)*. Association for Computational Linguistics, 2463–2473.

[16] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *OpenAI blog*, 12.

[17] Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: unanswerable questions for squad. In *ACL (2)*. Association for Computational Linguistics, 784–789.

[18] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100, 000+ questions for machine comprehension of text. In *EMNLP*. The Association for Computational Linguistics, 2383–2392.

[19] Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in bertology: what we know about how BERT works. *Trans. Assoc. Comput. Linguistics*, 8, 842–866.

[20] Sofia Serrano and Noah A. Smith. 2019. Is attention interpretable? In *ACL (1)*. Association for Computational Linguistics, 2931–2951.

[21] Wilson L Taylor. 1953. "cloze procedure": a new tool for measuring readability. *Journalism quarterly*, 30, 4, 415–433.

[22] Ian Tenney et al. 2019. What do you learn from context? probing for sentence structure in contextualized word representations. In *ICLR (Poster)*. OpenReview.net.

[23] Kristina Toutanova and Danqi Chen. 2015. Observed versus latent features for knowledge base and text inference. *Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality*, 57–66.

[24] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *ICML* (JMLR Workshop and Conference Proceedings). Vol. 48. JMLR.org, 2071–2080.

[25] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

[26] Jesse Vig. 2019. A multiscale visualization of attention in the transformer model. In *ACL (3)*. Association for Computational Linguistics, 37–42.

[27] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *BlackboxNLP@EMNLP*. Association for Computational Linguistics, 353–355.

[28] Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. 2017. Knowledge graph embedding: A survey of approaches and applications. *IEEE Trans. Knowl. Data Eng.*, 29, 12, 2724–2743.

[29] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph embedding by translating on hyperplanes. In *AAAI*. AAAI Press, 1112–1119.

[30] Sarah Wiegreffe and Yuval Pinter. 2019. Attention is not not explanation. In *EMNLP/IJCNLP (1)*. Association for Computational Linguistics, 11–20.

[31] Yonghui Wu et al. 2016. Google's neural machine translation system: bridging the gap between human and machine translation. *CoRR*, abs/1609.08144.

[32] Zhiyong Wu, Yun Chen, Ben Kao, and Qun Liu. 2020. Perturbed masking: parameter-free probing for analyzing and interpreting BERT. In *ACL*. Association for Computational Linguistics, 4166–4176.

[33] Patrick Xia, Shijie Wu, and Benjamin Van Durme. 2020. Which *bert? A survey organizing contextualized encoders. In *EMNLP (1)*. Association for Computational Linguistics, 7516–7533.

[34] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding entities and relations for learning and inference in knowledge bases. In *ICLR (Poster)*.

[35] Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. KG-BERT: BERT for knowledge graph completion. *CoRR*, abs/1909.03193.