Inspecting the concept knowledge graph encoded by modern language models

Carlos Aspillaga[1], **Marcelo Mendoza**[2], **Alvaro Soto**[1]

[1] Computer Science Department, Pontificia Universidad Católica de Chile
[2] Department of Informatics, Universidad Técnica Federico Santa María, Chile
`cjaspill@uc.cl,mmendoza@inf.utfsm.cl,asoto@ing.puc.cl`

Abstract

The field of natural language understanding has experienced exponential progress in the last few years, with impressive results in several tasks. This success has motivated researchers to study the underlying knowledge encoded by these models. Despite this, attempts to understand their semantic capabilities have not been successful, often leading to non-conclusive, or contradictory conclusions among different works. Via a probing classifier, we extract the underlying knowledge graph of nine of the most influential language models of the last years, including word embeddings, text generators, and context encoders. This probe is based on concept relatedness, grounded on WordNet. Our results reveal that all the models encode this knowledge, but suffer from several inaccuracies. Furthermore, we show that the different architectures and training strategies lead to different model biases. We conduct a systematic evaluation to discover specific factors that explain why some concepts are challenging. We hope our insights will motivate the development of models that capture concepts more precisely.

1 Introduction

Natural language processing (NLP) encompasses a wide variety of applications such as summarization (Kovaleva et al., 2019), information retrieval (Zhan et al., 2020), and machine translation (Tang et al., 2018), among others. Currently, the use of pre-trained language models has become the *de facto* starting point to tackle most of these tasks. The usual pipeline consists of finetuning a pre-trained language model by using a discriminative learning objective to adapt the model to the requirements of each task. As key ingredients, these models are pre-trained using massive amounts of unlabeled data that can include millions of documents and billions of parameters. Massive data and parameters are supplemented with a suitable learning architecture, resulting in a highly powerful but also complex model whose internal operation is hard to analyze.

The success of pre-trained language models has driven the interest to understand the mechanisms they use to solve NLP tasks. As an example, in the case of BERT (Devlin et al., 2019), one of the most popular pre-trained models based on the Transformer (Vaswani et al., 2017), several studies have attempted to access the knowledge encoded in its layers and attention heads (Tenney et al., 2019b; Devlin et al., 2019; Hewitt and Manning, 2019). In particular, Jawahar et al. (2019) shows that BERT can solve tasks at a syntactic level by using Transformer blocks to encode a soft hierarchy of features at different levels of abstraction. Similarly, Hewitt and Manning (2019) show that BERT is capable of encoding structural information from text. In particular, using a structural probe, they show that syntax trees are embedded in a linear transformation of the encodings of BERT.

In general, previous efforts have provided strong evidence indicating that current pre-trained language models encode complex syntactic rules. However, relevant evidence about their abilities to capture semantic information remains still elusive. As an example, Si et al. (2019) attempts to locate the encoding of semantic information as part of the top layers of Transformer architectures finding contradictory evidence. Similarly, Kovaleva et al. (2019) focuses on studying knowledge encoded by self-attention weights. Their results provide evidence for over-parameterization but not about language understanding capabilities.

In this work, we study to what extent pre-trained language models encode semantic information. As a key source of semantic knowledge, we analyze their ability to encode the concept relations embedded in the conceptual taxonomy of Word-

Net[1] (Miller, 1995). Understanding, organizing, and correctly using concepts is one of the most remarkable capabilities of human intelligence (Lake et al., 2017). Therefore, quantifying the ability that a pre-trained language model can exhibit to encode the conceptual organization behind WordNet is highly valuable. This knowledge may provide useful insights into the inner mechanisms that these models use to encode semantic information. Furthermore, identifying what they find difficult can provide relevant insights into how to improve them.

Unlike most previous works, we do not focus on a particular model but target a large list of the most popular pre-trained language models. In this sense, one of our goals is to provide a comparative analysis of the benefits of different approaches. Following Hewitt and Manning (2019), we study semantic performance by defining a probing classifier based on concept relatedness according to WordNet. Using this tool, we analyze the different models, enlightening how and where semantic knowledge is encoded. Furthermore, we explore how these models encode suitable information to recreate the structure of WordNet. Among our main results, we show that the different pre-training strategies and architectures lead to different model biases. In particular, we show that contextualized word embeddings, such as BERT, encode high-level concepts and hierarchical relationships among them, creating a taxonomy. This finding corroborates previous work results (Reif et al., 2019) that claim that BERT vectors store sub-spaces that correspond with semantic knowledge. Our study also shows evidence about the limitations of current pre-trained language models, demonstrating that they have difficulties to encode specific concepts. For example, all the models struggle with concepts related to "taxonomical groups". Our results also reveal that models have distinctive patterns regarding where in the architecture they encode the semantic information. These patterns are dependant on architecture and not on model sizes.

2 Study methodology

Probing methods consist of using the representation of a frozen pre-trained model to address a particular task. If the probing classifier succeeds in this setting but fails using an alternative model,

it means that the source model encodes the knowledge needed to solve the task. Furthermore, the classifier's performance can be used to measure how well the model captures this knowledge (Conneau et al., 2018). We use a probing method at the semantic level applying it to the nine models presented in Section 2.2. Our study sheds light on whether the models encode relevant knowledge to predict concept relatedness in Wordnet.

To study how accurately the models encode semantic information, we measure correctness in predicted relations among concepts at two levels: (a) pair-wise-level by studying performance across sampled pairs of related or unrelated concepts, and (b) graph-level by using pair-wise predictions to reconstruct the actual graph. We describe both approaches in Sections 2.3 and 2.4, respectively.

2.1 WordNet splits and sampling

We partitioned the available WordNet synsets at 70/15/15 for training, validation and test sets respectively. Our experimental setup ensures no overlap in concepts among these sets. As an example, if the concept related to "house" fell in the training set, then all its lemmas are considered in this partition (e.g. "home", "residence", etc.), and neither this concept nor those lemmas will be present in the validation or test sets. Our sampling setup also balances the number of times each concept acts as hypernym or as a hyponym in the relation, whenever possible. Thus the benefit of learning whether a word is a "prototypical hypernym", as pointed out by Levy et al. (2015), is close to zero. Further details are available in Appendix A.1.

2.2 Word embedding models

This study considers the most influential language models from recent years. We consider the essential approaches of three model families: non contextualized word embeddings (NCE), contextualized word embeddings (CE), and generative language models (GLM). We consider Word2Vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014) for the first family of approaches. For the CE family, we consider ELMo (Peters et al., 2018b), which is implemented on a bidirectional LSTM architecture, XLNet (Yang et al., 2019), and BERT (Devlin et al., 2019) and its extensions ALBERT (Lan et al., 2020) and RoBERTa (Liu et al., 2019), all of them based on the Transformer architecture. GPT-2 (Radford et al., 2018) and T5 (Raffel et al.,
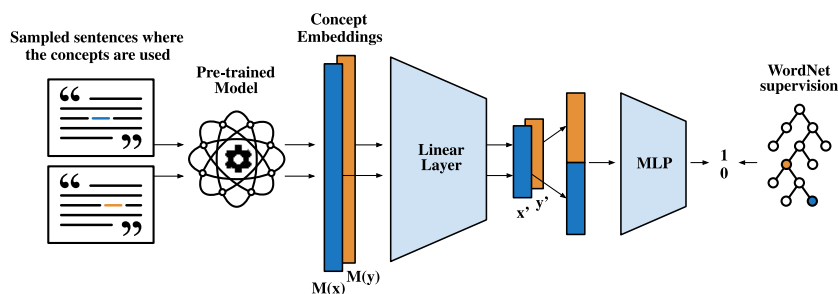
---

[1]WordNet is a human-generated graph, where each one of its 117000 nodes (also called synsets) represent a concept. In this work, we use hyponymy relations, representing if a concept is a subclass of another.

Figure 1: Inputs to the edge probing classifier correspond to the model embeddings $M(x)$ and $M(y)$ of concepts x and y, respectively. $M(x)$ and $M(y)$ are projected into a common lower dimensionality space using a linear layer. The resulting embeddings x' and y' are concatenated and fed into a Multi-Layer Perceptron that is in charge of predicting if the concept pair is related or not.

2019) are included in the study to incorporate approaches based on generative language models.

For models in the CE and GLM families, the embedding is extracted after running the model on a sentence where the concept is used in context. Then we discard the context and keep only the first token that correspond to the specific mention of the concept. Finally we concatenate the hidden states of every layer of the model, for the selected token.

2.3 Semantic probing classifier

We define an edge probing classifier that learns to identify if two concepts are semantically related. To create the probing classifier, we retrieve all the glosses from the Princeton WordNet Gloss Corpus[2]. This dataset provides WordNet's synsets gloss sentences with annotations identifying occurrences of concepts within different sentence contexts. The annotations provide a mapping of the used words to their corresponding WordNet node. We sample hypernym pairs A, B. Then, from an unrelated section of the taxonomy, we randomly sample a third synset C, taking care that C is not related to either A or B. Then, $\langle A, B, C \rangle$ forms a triplet that allows us to create six testing edges for our classifier. To train the probing classifier, we define a labeled edge $\{x, y, L\}$, with x and y synsets in $\{A, B, C\}$, $x \neq y$. $L \in \{0, 1\}$ is the target of the edge. If y is direct or indirect parent of x, $L = 1$, while $L = 0$ in other case. For each synset x, y, we sample one of its sentences $S(x)$, $S(y)$ from the dataset. Let M be a model. If M belongs to the NCE family, x and y are encoded by $M(x)$ and $M(y)$, respectively. If M belongs to the CE or GLM families, then x and y are encoded by the corresponding token of $M(S(x))$ and $M(S(y))$,

respectively.

To facilitate the evaluation of embeddings of different sizes, we first project each concept's encodings x and y into a low dimensionality space using a linear layer (see Figure 1). These vectors, denoted as x' and y', are concatenated and fed into a Multi-Layer Perceptron (MLP) classifier. The linear layer and the MLP are the only trainable parameters of our setting, as we use the source model weights without any finetuning. Throughout all the experiments we used an MLP classifier with a single hidden layer of 384 hidden units.

We use this MLP to learn the structural relation between concept pairs, providing the test with a mechanism that allows the embeddings to be combined in a non-linear way. Tests based on linear transformations such as the one proposed by Hewitt and Manning (2019) did not allow us to recover the WordNet structure. This indicates that the sub-spaces where the language models encode semantics are not linear. The fact that syntactic information is linearly available suggests that syntax trees might be a critical intermediate result for the language modeling task. In contrast, semantic information emerges as an indirect consequence of accurate language modeling. Still, it might not constitute information that the model relies on for NLP tasks, as postulated by Ravichander et al. (2020).

To discard the possibility of the MLP being memorizing properties of words and thus giving an undeserved credit to the analyzed models, we generated alternative training and validation sets with random word embeddings of the same size as the real ones. During training and inference, these vectors were kept frozen. These tests showed around 50% accuracy in the binary classification task, indicating that the MLP cannot do better than chance in that scenario. Thus, if in a later experiment the same MLP
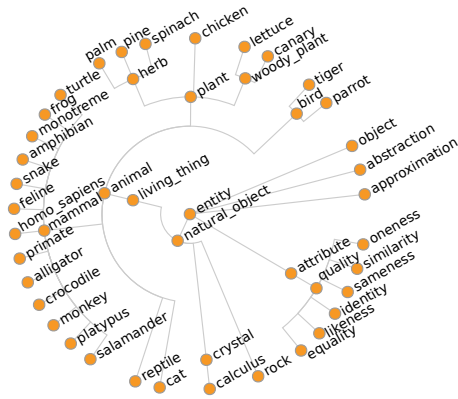
Figure 2: A reconstructed graph using BERT-large. Visual inspection reveals that the models capture key categories but fail to map fine-grained relations.

Family	Model	Tree Edit Dist.		
		TIM	**MCM**	**Avg.**
NCE	Word2Vec	59	59	**59**
	GloVe-42B	56	60	**58**
GLM	GPT-2	53	57	**55**
	T5	58	55	**56**
CE	ELMo	52	55	**53**
	BERT	49	48	**49**
	RoBERTa	56	54	**55**
	XLNet	52	48	**50**
	ALBERT	53	50	**51**

Table 1: Tree Edit Distance against the ground truth graph (large models used). We display both strategies for estimating d_e along with their average score.

succeeds at the task, the merit can be attributed to the input embedding itself. This result is consistent with the fact that our experimental setup ensures no overlap in concepts among training, development, and testing sets.

2.4 Reconstructing the structure of a knowledge graph

The probe classifier predicts if a pair of concepts $\langle u, v \rangle$ form a valid $\langle \text{parent}, \text{child} \rangle$ relation according to WordNet, where $h_{\langle u,v \rangle} \in [0,1]$ denotes the corresponding classifier output. It is important to note that valid $\langle \text{parent}, \text{child} \rangle$ relations include direct relations (e.g. $\langle \text{dog}, \text{poodle} \rangle$), and transitive relations (e.g. $\langle \text{animal}, \text{poodle} \rangle$), and that the order of the items matters.

To reconstruct the underlying knowledge graph, for each valid $\langle \text{parent}, \text{child} \rangle$ relation given by $h_{\langle u,v \rangle} >$ threshold, we need an estimation of how close are the nodes in the graph. We do this by introducing the concept of "parent closeness" between a parent node u and a child node v, denoted by $d_e(u, v)$. We propose two alternative scores to estimate d_e:

i) Model Confidence Metric (MCM): All the models considered in this study capture close relations more precisely than distant relations (supporting evidence can be found in Appendix D). This means that a concept like *poodle* will be matched with its direct parent node *dog* with higher confidence than with a more distant parent node (e.g. *animal*). Thus, we can define $d_e(u, v) = 1 - h_{\langle u,v \rangle}$.

ii) Transitive Intersections Metric (TIM): We explore a metric grounded directly in the tree structure of a knowledge graph. Note that nodes u and

v that form a parent-child relation have some transitive connections in common. Specifically, all descendants of v are also descendants of u, and all the ancestors of u are also ancestors of v. Then, the closer the link between u and v in the graph, the bigger the intersection. Accordingly, for each edge $e = \langle u, v \rangle$, we define $d_e(u, v)$ as:

$$-\left(\sum_{j \in N \setminus \{u,v\}} h_{\langle u,j \rangle} h_{\langle v,j \rangle} + h_{\langle j,u \rangle} h_{\langle j,v \rangle} \right) * h_{\langle u,v \rangle}, \tag{1}$$

where the first term of the sum accounts for the similarity within the descendants of nodes u and v, and the second term accounts for the similarity within the ancestors of nodes u and v. The term $h_{\langle u,v \rangle}$ at the right-hand side accounts for the edge direction, and N denotes the set of nodes (concepts).

A strategy to find a tree that comprises each node's closest parents is the minimum-spanning-arborescence (MSA) of the graph defined using d_e. The MSA is analogous to the minimum-spanning-tree (MST) objective used by Hewitt and Manning (2019), but for directed graphs. The formulation of the MSA optimization problem applied to our proposal is provided in the Appendix A.3.

3 How accurate is this knowledge?

3.1 Semantic edge probing classifier results

Table 2 shows the results obtained using the edge probing classifier. Results show that regardless of model sizes, performance is homogeneous within each family of models. Additionally, results show that NCE and GLM methods obtain a worse performance when all the layers are used than those achieved by CE methods. When single layers are

Family	Model	Emb. Size All/Best Layer	Best Layer	F1-score All Layers	F1-score Best Layer
NCE	Word2Vec (Mikolov et al., 2013)	300 / -	-	.7683 ± .0135	-
	GloVe-42B (Pennington et al., 2014)	300 / -	-	.7877 ± .0084	-
GLM	GPT-2 (Radford et al., 2018)	9984 / 768	6	.7862 ± .0132	.7921 ± .0108
	T5-small (Raffel et al., 2019)	7168 / 512	4	.8156 ± .0098	.8199 ± .0081
	GPT2-xl (Radford et al., 2018)	78400 / 1600	13	.7946 ± .0151	.8029 ± .0118
	T5-large (Raffel et al., 2019)	51200 / 1024	17	.8148 ± .0119	.8331 ± .0102
CE	ELMo-small (Peters et al., 2018b)	768 / 256	2	.7986 ± .0126	.7880 ± .0119
	BERT-base (Devlin et al., 2019)	9984 / 768	10	.8240 ± .0123	.8185 ± .0104
	RoBERTa-base (Liu et al., 2019)	9984 / 768	5	.8392 ± .0100	.8266 ± .0083
	XLNet-base (Yang et al., 2019)	9984 / 768	4	.8306 ± .0113	.8293 ± .0116
	ALBERT-base (Lan et al., 2020)	9984 / 768	12	.8184 ± .0222	.8073 ± .0102
	ELMo-large (Peters et al., 2018b)	3072 / 1024	2	.8311 ± .0090	.8330 ± .0083
	BERT-large (Devlin et al., 2019)	25600 / 1024	14	.8178 ± .0152	.8185 ± .0113
	RoBERTa-large (Liu et al., 2019)	25600 / 1024	13	.8219 ± .0159	.8314 ± .0082
	XLNet-large (Yang et al., 2019)	25600 / 1024	6	.8211 ± .0142	.8244 ± .0080
	ALBERT-xxlarge (Lan et al., 2020)	53248 / 4096	4	.8233 ± .0107	.8194 ± .0097

Table 2: Results obtained using the edge probing classifier. We study the performance in many model variants, considering small and large versions of several models. Results are grouped by method families.

used, GLM shows improved performance, suggesting that these models capture semantics earlier in the architecture, keeping their last layers for generative-specific purposes. In contrast, CE models degrade or maintain their performance when single layers are used.

Note that Table 2 shows pair-wise metrics not graph metrics. As we are dealing with graphs, predicted edges are built upon related edges. Thus, drifts in small regions of the graph may cause large drifts in downstream connections. Furthermore, our setup balances positive and negative samples. However, the proportion of negative samples can be considerably larger in a real reconstruction scenario. As a consequence, we emphasize that these numbers must be considered together with the results reported in sections 3.2 and 4.

3.2 Extracting the Knowledge Graph

Predicting a knowledge graph has a complexity of at least $O(N^2)$ in the number of analyzed concepts. In our case, this imposes a highly demanding computational obstacle because WordNet has over 82000 noun synsets. To accelerate experimentation and facilitate our analysis and visualizations, we focus on extracting a WordNet sub-graph comprising 46 nodes not seen during training or validation. These nodes are picked to include easily recognizable relations. We use the tree-edit-distance to evaluate how close are the reconstructed graphs to

the target graph extracted from WordNet. Table 1 shows our results.

Table 1 shows that graphs retrieved using CE models are closer to the target than graphs provided by NCE and GLM models. In particular, the best results are achieved by BERT, ALBERT, and XLNet, indicating that these models encode more accurate semantic information than the alternative models. These results are consistent with those obtained in Section 3.1. The graphs for all the models can be found in Appendix C.

4 What is easy or hard? What are these models learning?

Section 3 shows that different model families differ in their errors. Furthermore, it shows that within the same family, models have similar biases. In this section, we elucidate which semantic factors impact the performance of these models and which ones do not affect their F1-score.

Figure 3-a shows that most models decrease their F1-score as concepts get more specific. We hypothesize that higher-level concepts (e.g., *Animal*) appear more frequently and in more diverse contexts, as they are also seen as instances of their sub-classes (e.g., *Dog*, *Cat*, *Chihuahua*), allowing the models to learn more precise representations for them. In contrast, lower-level concepts will only appear in specific contexts (e.g., texts about *Apple-Head-Chihuahua*). Figure 3-b corroborates
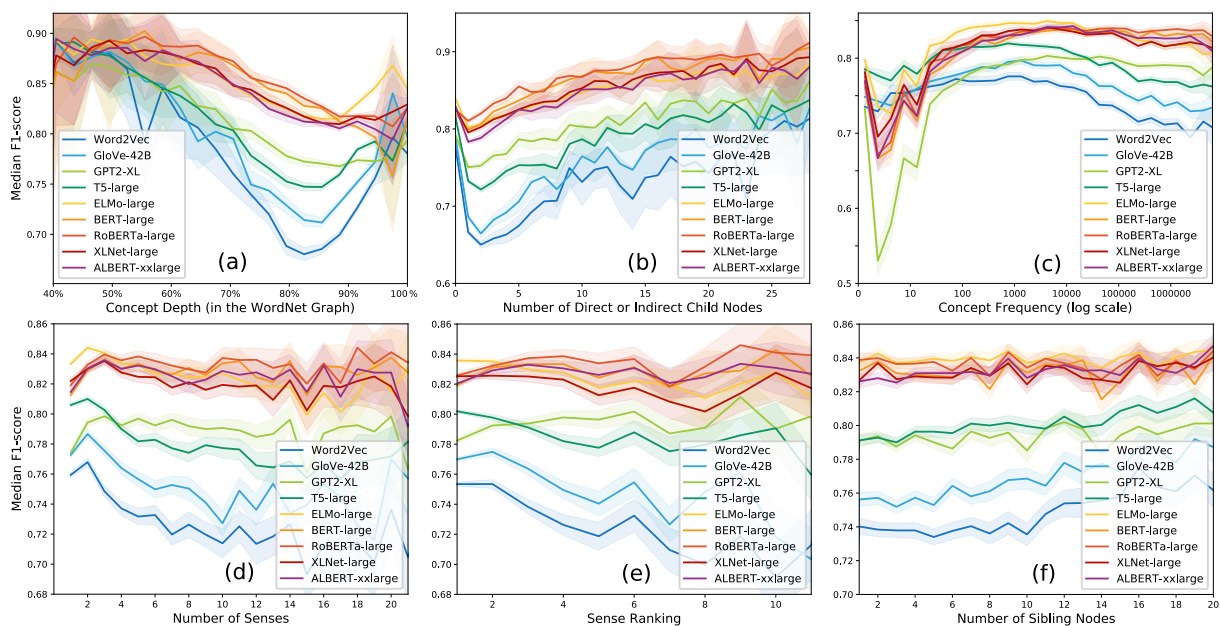
Figure 3: Semantic factors with a high (top charts) or low (bottom charts) impact on F1-score, along with their 90% confidence intervals. Charts only display ranges where at least 100 samples existed. Appendix D shows additional factors along with the specific implementation details.

this intuition, as concepts with a higher number of sub-classes have higher F1-scores. Figure 3-c shows that models degrade their F1-score when concepts are too frequent. In particular, NCE and GLM models are more sensitive to this factor.

Another finding is that CE and GLM models are almost unaffected by the number of senses that a certain word has, neither to their sense ranking or their number of sibling concepts, displaying almost flat charts (see Figures 3-d-e-f). This result suggests that these models pay more attention to the context than to the target word. This behavior is opposed to what NCE models exhibit according to Yaghoobzadeh et al. (2019), as NCE models tend to focus more on frequent senses.

In most cases, the same family models have similar behaviors, especially within the NCE or CE families. Also, different families show different patterns. Table 3 shows some salient examples. Surprisingly, all models struggle in the category "taxonomic groups". Manual inspection of sentences makes us believe that the context confuses CE and GLM models in these cases. In many sentences, the corresponding concept could be nicely replaced by another, conveying a modified but still valid message. This phenomenon does not occur in other categories such as "social group" or "attribute", even though these concepts are closely related to "taxonomic groups".

5   Where is this knowledge located?

As mentioned in Section 7, prior work has not shown consensus about where is semantic information encoded inside these architectures. Our experiments shed light on this subject. Figure 4 shows how each layer contributes to the F1-score.

Figures 4-a and 4-b show the performance across layers for the CE-based models. They reveal that while BERT and RoBERTa use their top-layers to encode semantic information, XLNet and AL-BERT use the first layers. Figure 4-c shows that while GPT-2 uses all its layers to encode semantics, T5 shows an M shape related to its encoder-decoder architecture. The chart shows that T5 uses its encoder to hold most of the semantic information. We also note that small models show similar patterns as their larger counterparts.

6   Further discussion and implications

Table 4 summarizes our main findings. Findings (1), (2), and (3) indicate that, to a different extent, all models encode relevant knowledge about the hierarchical semantic relations included in Word-Net. However, as we mention in Section 4, we observe that the ability to learn about a concept depends on its frequency in the training corpus and the specificity of its meaning. Furthermore, some concept categories seem to be hard for every model family, while some are particularly difficult

Family	Model	artifact	attribute	living thing	matter	person	relation	part	social group	taxonomic group
NCE	Word2Vec	.7120	.7044	.7295	.7402	.7208	.7264	.7532	.7497	.6920
	GloVe-42B	.7389	.7213	.7421	.7633	.7351	.7567	.7759	.7579	.6648
GLM	GPT-2	.7903	.7730	.7300	.7582	.7207	.7612	.7540	.8155	.3030
	T5	.7868	.7649	.7862	.8002	.7735	.7963	.8051	.7868	.6944
CE	ELMo	.8308	.8093	.8187	.7756	.8022	.7679	.7580	.8312	.6011
	BERT	.8249	.8094	.7593	.7645	.7379	.7662	.7499	.8516	.4804
	RoBERTa	.8315	.8167	.7823	.7614	.7585	.7649	.7441	.8552	.4921
	XLNet	.8319	.8064	.7907	.7636	.7779	.7659	.7526	.8422	.5371
	ALBERT	.8231	.8050	.7758	.7685	.7826	.7727	.7610	.8556	.4277

Table 3: Average F1-score for some semantic categories revealing models strengths and weaknesses. Several other categories are reported in Appendix E along with their standard deviations.
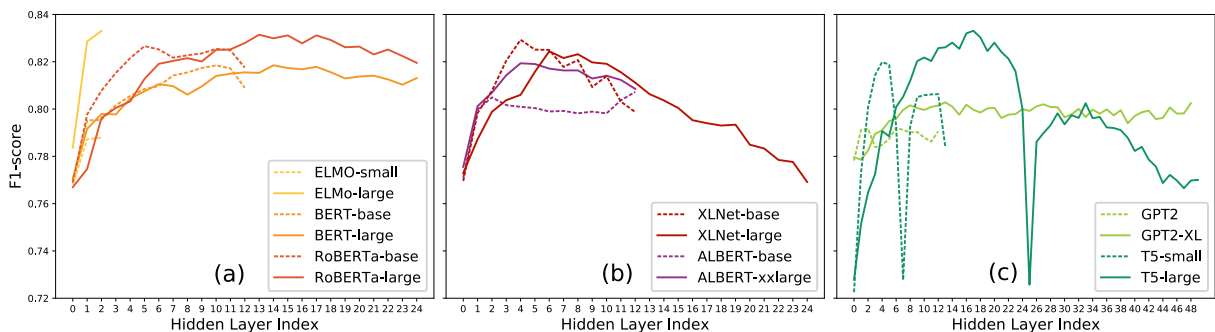

Figure 4: F1-score for hypernym prediction across each model layer.

for contextual models such as CE. We hypothesize that stronger inductive biases are required to capture low-frequency concepts. Furthermore, we believe that new learning approaches are needed to discriminate accurate meaning for high-frequency concepts. As expected, our findings indicate that model families have different biases leading to different behaviors. Thus, our results can illuminate further research to improve semantic capabilities by combining each family of models' strengths. For example, one could combine them as ensembles, each one equipped with a different loss function (i.e., one generative approach resembling GLM-based methods and another discriminative resembling CE-based methods).

Findings (4), (5), and (6) suggest that instead of a standard finetuning of all layers of BERT according to a given downstream task, to improve semantic capabilities, one could perform a task profiling to decide the best architecture for the task and also how to take advantage of it. Using only a limited number of layers or choosing a different learning rate for each layer, one could exploit the semantic knowledge that the pre-trained model carries, avoiding the degradation of this information present at the top layers, especially when using

T5, XLNet, or ALBERT-large. Accordingly, recent work on adaptive strategies to output predictions using a limited number of layers (Xin et al., 2020; Liu et al., 2020; Hou et al., 2020; Schwartz et al., 2020; Fan et al., 2020; Bapna et al., 2020) would benefit from using architectures that encode knowledge in the first layers. To the best of our knowledge, these works have only used BERT and RoBERTa, achieving a good trade-off between accuracy and efficiency. Only Zhou et al. (2020) has explored ALBERT, reporting improved accuracy by stopping earlier. Our findings explain this behavior and suggest that T5 or XLNet may boot their results even further as these architectures have sharper and higher information peaks in their first layers.

Findings (7) and (8) suggest that recent success in semantic NLP tasks might be due more to the use of larger models than large corpora for pretraining. This also suggests that to improve model performance in semantic tasks, one could train larger models even without increasing the corpus size. A similar claim has been proposed by (Li et al., 2020) leading to empirical performance improvements.

Finally, finding (9) is important because it suggests that contextual models pay as much attention to the context as to the target word and are probably

	Finding	Supporting Evidence	Involved Models
(1)	All models encode a relevant amount of knowledge about semantic relations in WordNet, but this knowledge contains imprecisions.	All	All
(2)	The ability to learn concept relations depends on how frequent and specific the concepts are. Some model families are more affected.	Fig. 3a-c	NCE and GLM
(3)	Concept difficulty is usually homogeneous within each model family. Some semantic categories challenge all models.	Table 3	All
(4)	Some models encode stronger semantic knowledge than others, usually according to their family.	Tables 2, 1, 3	ELMo, BERT, RoBERTa, ALBERT, XLNet, T5
(5)	Some models focus their encoding of semantic knowledge in specific layers, and not distributed across all layers.	Table 2, Fig. 4	GLM
(6)	Models have distinctive patterns as to where they encode semantic knowledge. Patterns are model-specific and not size-specific.	Table 2, Fig. 4	All
(7)	Model size has an impact in the quality of the captured semantic knowledge, as seen in our layer-level probe tests.	Table 2, Fig. 4	ELMo, RoBERTa, ALBERT, GPT-2, T5
(8)	Semantic knowledge does not depend on pre-training corpus size.	Tables 2, B-5	-
(9)	Contextual models are unaffected by multi-sense words.	Fig. 3d-f	CE and GLM

Table 4: Summary of our main findings and their corresponding supporting evidence.

biased in favor of contextual information, even if they are not based on the Masked-Language-Model strategy. We believe that this inductive bias could be exploited even further in the design of the underlying architecture. Thus this finding might elucidate a design direction to encourage more effective learning of semantic knowledge.

7 Related work

The success of deep learning architectures in various NLP tasks has fueled a growing interest to improve understanding of what these models encode. Studies like Tenney et al. (2019b) claim that success in a specific task helps understand what type of information the model encodes.

Evidence of syntactic information: Using probing classifiers, Clark et al. (2019) claims that some specific BERT's attention heads show correspondence with syntactic tasks. Goldberg (2019) illustrates the capabilities that BERT has to solve syntactic tasks, such as subject-verb agreement. Hewitt and Manning (2019) proposes a structural probe that evaluates whether syntax trees are encoded in a linear transformation of BERT embeddings. The study provides evidence that syntax trees are implicitly embedded in BERT's vector geometry. Reif et al. (2019) has found evidence of syntactic representation in BERT's attention matrices, with specific directions in space representing particular dependency relations.

Evidence of semantic information: Reif et al. (2019) suggests that BERT's internal geometry may be broken into multiple linear subspaces, with separate spaces for different syntactic and semantic information. Despite this result, previous work has not yet reached a consensus about this topic. While some studies show satisfactory results in tasks such as entity types (Tenney et al., 2019a), semantic roles (Rogers et al., 2020), and sentence completion (Ettinger, 2020), other studies show less favorable results in coreference (Tenney et al., 2019b), Multiple-Choice Reading Comprehension (Si et al., 2019) and Lexical Relation Inference (Levy et al., 2015), claiming that BERT's performance may not reflect the model's true ability of language understanding and reasoning. Tenney et al. (2019b) proposes a set of edge probing tasks to test the encoded sentential structure of contextualized word embeddings. The study shows evidence that the improvements that BERT and GPT-2 offer over non contextualized embeddings as GloVe is only significant in syntactic-level tasks. Regarding static word embeddings, Yaghoobzadeh et al. (2019) shows that senses are well represented in single-vector embeddings if they are frequent and that this does not harm NLP tasks whose performance depends on frequent senses.

Layer-wise or head-wise information: Tenney et al. (2019a) shows that the first layers of BERT focus on encoding short dependency relationships at the syntactic level (e.g., subject-verb agreement). In contrast, top layers focus on encoding long-

range dependencies (e.g., subject-object dependencies). Peters et al. (2018a) supports similar declarations for Convolutional, LSTM, and self-attention architectures. While these studies also support that the top layers appear to encode semantic information, the evidence to support this claim is not conclusive or contradictory with other works. For example, Jawahar et al. (2019) could only identify one SentEval semantic task that topped at the last layer. In terms of information flow, Voita et al. (2019a) reports that information about the past in left-to-right language models gets vanished as the information flows from bottom to top BERT's layers. Hao et al. (2019) shows that the lower layers of BERT change less during finetuning, suggesting that layers close to inputs learn more transferable language representations. Press et al. (2020) shows that increasing self-attention at the bottom layers improves language modeling performance based on BERT. Other studies focus on understanding how self-attention heads contribute to solving specific tasks (Vig, 2019). Kovaleva et al. (2019) shows a set of attention patterns repeated across different heads when trying to solve GLUE tasks (Wang et al., 2018). Furthermore, Michel et al. (2019) and Voita et al. (2019b) show that several heads can be removed without harming downstream tasks.

Automated extraction of concept relations: Although the main focus of our work is not to master the probing task of extracting knowledge from WordNet, but to use it as an instrument to verify and compare the abilities of current families of language models to encode this kind of knowledge, for completitude we include a brief mention of previous literature regarding this subject. Relation extraction is an active research topic. Early works are either feature-based, usually relying on SVMs, Maximum Entropy, or on a set of manually defined rules (Hearst, 1998; Kambhatla, 2004; Dashtipour et al., 2017; Minard et al., 2011; Weeds et al., 2014; Chen et al., 2015). Other methods rely on manually defined distance metrics to estimate the relatedness of two semantic instances (Dandan et al., 2012; Panyam et al., 2016). Following works have used different types of neural networks or LSTM modules for this task (Liu et al., 2013; Zeng et al., 2014, 2015; Zhang and Wang, 2015; Song et al., 2018), or attention-based and transformer-based mechanisms with outstanding results (Zhou et al., 2016; Baldini Soares et al., 2019; Huang et al.,

2020; Qin et al., 2021; Zhong and Chen, 2021).

Alternative approaches: Several alternative approaches have been used in previous works. Some are dataset-focused (Miller et al., 1994; Levy et al., 2015; Wang et al., 2018; Wiedemann et al., 2019), usually relying on annotated corpora that challenge semantic abilities. These approaches have provided useful insights, but usually suffer from low availability of data as they usually cover a small fraction of the WordNet ontology. As an example, BLESS (Baroni and Lenci, 2011) includes gold-standard annotations for only 200 concepts. Other approaches have tested semantic ability by using prompt-engineering and inspecting the predictions of the models (Petroni et al., 2019; Ettinger, 2020; Talmor et al., 2020), but other works have also shown a high variability in the results depending on the prompt design (Balasubramanian et al., 2020; Reynolds and McDonell, 2021; Zhao et al., 2021).

8 Conclusions

In this work, we exploit the semantic conceptual taxonomy behind WordNet to test the ability of current families of pre-trained language models to learn semantic knowledge from massive sources of unlabeled data. Our main conclusion is that, indeed, to a significant extent, these models learn relevant knowledge about the organization of concepts in WordNet, but also contain several imprecisions. We also notice that different families of models present dissimilar behavior, suggesting the encoding of different biases.

We hope our study helps to inspire new ideas to improve the semantic learning abilities of current pre-trained language models.

Acknowledgments

References

Sriram Balasubramanian, Naman Jain, Gaurav Jindal, Abhijeet Awasthi, and Sunita Sarawagi. 2020. What's in a name? are BERT named entity representations just as good for any other name? In *Proceedings of the 5th Workshop on Representation Learning for NLP*, pages 205–214, Online. Association for Computational Linguistics.

Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. Matching the blanks: Distributional similarity for relation learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2895–2905, Florence, Italy. Association for Computational Linguistics.

Ankur Bapna, Naveen Arivazhagan, and Orhan Firat. 2020. Controlling computation versus quality for neural sequence models. *CoRR*, abs/2002.07106.

Marco Baroni and Alessandro Lenci. 2011. How we BLESSed distributional semantic evaluation. In *Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics*, pages 1–10, Edinburgh, UK. Association for Computational Linguistics.

Yanping Chen, Qinghua Zheng, and Ping Chen. 2015. Feature assembly method for extracting relations in chinese. *Artificial Intelligence*, 228:179–194.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does BERT look at? An analysis of BERT's attention. *CoRR*, abs/1906.04341.

Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136, Melbourne, Australia. Association for Computational Linguistics.

Liu Dandan, Hu Yanan, and Qian Longhua. 2012. Exploiting lexical semantic resource for tree kernel-based chinese relation extraction. In *Natural Language Processing and Chinese Computing. NLPCC 2012. Communications in Computer and Information Science, vol 333. Springer, Berlin, Heidelberg*.

Kia Dashtipour, Mandar Gogate, Ahsan Adeel, Abdulrahman Algarafi, Newton Howard, and Amir Hussain. 2017. Persian named entity recognition. In *2017 IEEE 16th International Conference on Cognitive Informatics Cognitive Computing (ICCI*CC)*, pages 79–83.

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, NAACL-HLT 2019*, pages 4171–4186.

Allyson Ettinger. 2020. What BERT is not: Lessons from a new suite of psycholinguistic diagnostics for language models. *Transactions of the. Association of Computational Linguistics, TACL*, 8:34–48.

Angela Fan, Edouard Grave, and Armand Joulin. 2020. Reducing transformer depth on demand with structured dropout. In *International Conference on Learning Representations*.

Yoav Goldberg. 2019. Assessing BERT's syntactic abilities. *CoRR*, abs/1901.05287.

Yaru Hao, Li Dong, Furu Wei, and Ke Xu. 2019. Visualizing and understanding the effectiveness of BERT. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019*, pages 4141–4150.

Marti A. Hearst. 1998. Automated discovery of wordnet relations. In *WordNet: An Electronic Lexical Database, Christiane Fellbaum (ed.), MIT Press*.

John Hewitt and Christopher D. Manning. 2019. 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, NAACL-HLT 2019*, pages 4129–4138.

Lu Hou, Zhiqi Huang, Lifeng Shang, Xin Jiang, Xiao Chen, and Qun Liu. 2020. Dynabert: Dynamic bert with adaptive width and depth. In *Advances in Neural Information Processing Systems*.

Wenti Huang, Yiyu Mao, Zhan Yang, Lei Zhu, and Jun Long. 2020. Relation classification via knowledge graph enhanced transformer encoder. *Knowledge-Based Systems*, 206:106321.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019*, pages 3651–3657.

Nanda Kambhatla. 2004. Combining lexical, syntactic, and semantic features with maximum entropy models for information extraction. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pages 178–181, Barcelona, Spain. Association for Computational Linguistics.

Olga Kovaleva, Alexey Romanov, Anna Rogers, and Anna Rumshisky. 2019. Revealing the dark secrets of BERT. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019*, pages 4365–4374.

Brenden M. Lake, Tomer D. Ullman, Joshua B. Tenenbaum, and Samuel J. Gershman. 2017. Building machines that learn and think like people. *Behavioral and brain sciences*, 40.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *8th International Conference on Learning Representations, ICLR 2020*.

Omer Levy, Steffen Remus, Chris Biemann, and Ido Dagan. 2015. Do supervised distributional methods really learn lexical inference relations? In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 970–976, Denver, Colorado. Association for Computational Linguistics.

Zhuohan Li, Eric Wallace, Sheng Shen, Kevin Lin, K. Keutzer, D. Klein, and J. Gonzalez. 2020. Train large, then compress: Rethinking model size for efficient training and inference of transformers. *ArXiv*, abs/2002.11794.

Chunyang Liu, Wenbo Sun, Wenhan Chao, and Wanxiang Che. 2013. Convolution neural network for relation extraction. In *Part II of the Proceedings of the 9th International Conference on Advanced Data Mining and Applications - Volume 8347*, ADMA 2013, page 231–242, Berlin, Heidelberg. Springer-Verlag.

Weijie Liu, Peng Zhou, Zhiruo Wang, Zhe Zhao, Haotang Deng, and Qi Ju. 2020. FastBERT: a self-distilling BERT with adaptive inference time. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6035–6044, Online. 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. *CoRR*, abs/1907.11692.

Paul Michel, Omer Levy, and Graham Neubig. 2019. Are sixteen heads really better than one? In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019*, pages 14014–14024.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26, NIPS 2013*, pages 3111–3119.

George A. Miller. 1995. WordNet: A lexical database for english. *Communications of the ACM*, 38(11):39–41.

George A. Miller, Martin Chodorow, Shari Landes, Claudia Leacock, and Robert G. Thomas. 1994. Using a semantic concordance for sense identification. In *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*.

Anne-Lyse Minard, Anne-Laure Ligozat, and Brigitte Grau. 2011. Multi-class SVM for relation extraction from clinical reports. In *Proceedings of the International Conference Recent Advances in Natural Language Processing 2011*, pages 604–609, Hissar, Bulgaria. Association for Computational Linguistics.

Nagesh C. Panyam, Karin Verspoor, Trevor Cohn, and Rao Kotagiri. 2016. ASM kernel: Graph kernel using approximate subgraph matching for relation extraction. In *Proceedings of the Australasian Language Technology Association Workshop 2016*, pages 65–73, Melbourne, Australia.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014*, pages 1532–1543.

Matthew Peters, Mark Neumann, Luke Zettlemoyer, and Wen-tau Yih. 2018a. Dissecting contextual word embeddings: Architecture and representation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1499–1509, Brussels, Belgium. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018b. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018*, pages 2227–2237.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Ofir Press, Noah A. Smith, and Omer Levy. 2020. Improving transformer models by reordering their sublayers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020*, pages 2996–3005.

Yongbin Qin, Weizhe Yang, Kai Wang, Ruizhang Huang, Feng Tian, Shaolin Ao, and Yanping Chen. 2021. Entity relation extraction based on entity indicators. *Symmetry*, 13(4).

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2018. Language models are unsupervised multitask learners.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:1–67.

Abhilasha Ravichander, Yonatan Belinkov, and E. Hovy. 2020. Probing the probing paradigm: Does probing accuracy entail task relevance? *ArXiv*, abs/2005.00719.

Emily Reif, Ann Yuan, Martin Wattenberg, Fernanda B. Viégas, Andy Coenen, Adam Pearce, and Been Kim. 2019. Visualizing and measuring the geometry of BERT. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019*, pages 8592–8600.

Laria Reynolds and Kyle McDonell. 2021. *Prompt Programming for Large Language Models: Beyond the Few-Shot Paradigm*. Association for Computing Machinery, New York, NY, USA.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in BERTology: What we know about how BERT works. CoRR abs/2002.12327.

Roy Schwartz, Gabriel Stanovsky, Swabha Swayamdipta, Jesse Dodge, and Noah A. Smith. 2020. The right tool for the job: Matching model and instance complexities. In *Proc. of ACL*.

Chenglei Si, Shuohang Wang, Min-Yen Kan, and Jing Jiang. 2019. What does BERT learn from multiple-choice reading comprehension datasets? *CoRR*, abs/1910.12391.

Linfeng Song, Yue Zhang, Zhiguo Wang, and Daniel Gildea. 2018. N-ary relation extraction using graph-state LSTM. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2226–2235, Brussels, Belgium. Association for Computational Linguistics.

Alon Talmor, Yanai Elazar, Yoav Goldberg, and Jonathan Berant. 2020. oLMpics-on what language model pre-training captures. *Transactions of the Association for Computational Linguistics*, 8:743–758.

Gongbo Tang, Rico Sennrich, and Joakim Nivre. 2018. An analysis of attention mechanisms: The case of word sense disambiguation in neural machine translation. In *Proceedings of the Third Conference on Machine Translation, WMT 2018*, pages 26–35.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019a. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019*, pages 4593–4601.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R. Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R. Bowman, Dipanjan Das, and Ellie Pavlick. 2019b. What do you learn from context? probing for sentence structure in contextualized word representations. In *7th International Conference on Learning Representations, ICLR 2019*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems NIPS 2017*, pages 5998–6008.

Jesse Vig. 2019. A multiscale visualization of attention in the transformer model. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019*, pages 37–42.

Elena Voita, Rico Sennrich, and Ivan Titov. 2019a. The bottom-up evolution of representations in the transformer: A study with machine translation and language modeling objectives. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019*, pages 4395–4405.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. 2019b. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019*, pages 5797–5808.

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 *Proceedings of the Workshop: Analyzing and Interpreting Neural Networks for NLP, BlackboxNLP@EMNLP 2018*, pages 353–355.

Julie Weeds, Daoud Clarke, Jeremy Reffin, David Weir, and Bill Keller. 2014. Learning to distinguish hypernyms and co-hyponyms. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2249–2259, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Gregor Wiedemann, Steffen Remus, Avi Chawla, and Chris Biemann. 2019. Does BERT make any sense? interpretable word sense disambiguation with contextualized embeddings.

Ji Xin, Raphael Tang, Jaejun Lee, Yaoliang Yu, and Jimmy Lin. 2020. DeeBERT: Dynamic early exiting for accelerating BERT inference. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2246–2251, Online. Association for Computational Linguistics.

Yadollah Yaghoobzadeh, Katharina Kann, Timothy J. Hazen, Eneko Agirre, and Hinrich Schütze. 2019. Probing for semantic classes: Diagnosing the meaning content of word embeddings. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 5740–5753. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. XLNet: Generalized autoregressive pretraining for language understanding. In *Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019*, pages 5754–5764.

Daojian Zeng, Kang Liu, Yubo Chen, and Jun Zhao. 2015. Distant supervision for relation extraction via piecewise convolutional neural networks. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1753–1762, Lisbon, Portugal. Association for Computational Linguistics.

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. 2014. Relation classification via convolutional deep neural network. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2335–2344, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Min Zhang, and Shaoping Ma. 2020. An analysis of BERT in document ranking. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020*, pages 1941–1944.

Dongxu Zhang and Dong Wang. 2015. Relation classification via recurrent neural network.

Tony Z. Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. *arXiv preprint arXiv:2102.09690*.

Zexuan Zhong and Danqi Chen. 2021. A frustratingly easy approach for entity and relation extraction. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 50–61, Online. Association for Computational Linguistics.

Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. 2016. Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 207–212, Berlin, Germany. Association for Computational Linguistics.

Wangchunshu Zhou, Canwen Xu, Tao Ge, Julian McAuley, Ke Xu, and Furu Wei. 2020. Bert loses patience: Fast and robust inference with early exit.

A Implementation details

A.1 Edge probing classifier details

To study the extent to which these Language Models deal with semantic knowledge, we extend the methodology introduced by Tenney et al. (2019b). In that study, the authors defined a probing classifier at the sentence level, training a supervised classifier with a task-specific label. The probing classifier's motivation consists of verifying when the sentence's encoding help to solve a specific task, quantifying these results for different word embeddings models. We cast this methodology to deal with semantic knowledge extracted from WordNet. Rather than working at the sentence level, we define an edge probing classifier that learns to identify if two concepts are semantically related.

To create the probing classifier, we retrieve all the glosses from the Princeton WordNet Gloss Corpus. The dataset provides WordNet's synsets gloss with manually matched words identifying the context-appropriate sense.

As a reference of size, the selected annotations in the corpus accounted for 41502 lemmas, corresponding to 34371 WordNet synsets. This resulted in 230215 valid WordNet relations.

In WordNet, each sense is coded as one of the synsets related to the concept (e.g., sense *tendency.n.03* for the word tendency). Using a synset A and its specific sense provided by the tagged gloss, we retrieve from WordNet one of its direct or indirect hypernyms, denoted as B (see Figure 5). If WordNet defines two or more hypernyms for A, we choose one of them at random. We sample a third synset C, at random from an unrelated section of the taxonomy, taking care that C is not related to either A or B (e.g., *animal.n.01*). Then, $\langle A, B, C \rangle$ form a triplet that allows us to create six testing edges for our classifier: $\langle A, B \rangle$, which is compounded by a pair of related words through the semantic relation *hypernym of*, and five pairs of unrelated words ($\langle A, C \rangle$, $\langle B, C \rangle$, $\langle B, A \rangle$, $\langle C, A \rangle$, $\langle C, B \rangle$). We associate a label to each of these pairs that show whether the pair is related or not (see Figure 5). Note that we define directed edges, meaning that the pair $\langle A, B \rangle$ is related, but $\langle B, A \rangle$ is unrelated to the relationship *hypernym of*. Accordingly, the edge probing classifier will need to identify the pair's components and the order in which the concepts were declared in the pair.

We create training and testing partitions ensuring that each partition has the same proportion of leaves versus internal nodes. The latter is essential to identify related pairs. During training, we guarantee that each training synset is seen at least once by the probing classifier. To guarantee the above, we sample each synset in the training set and sample some of its hypernyms at random. Then. we randomly sample some unrelated synset for each related pair that has no relation to any of the words in the related pair. We create three partitions from this data on 70/15/15 for training, development, and testing foldings, respectively.

We train the MLP classifier using a weighted binary cross-entropy loss function. Since we have one positive and five negative examples per triplet, we use a weighted loss function with weights 5 and 1 for the positive and negative class, respectively. Accordingly, positive and negative examples have the same relevance during training. We implemented the linear layer and the MLP classifier using a feed forward network with 384 hidden units. The MLP was trained using dropout at 0.425 and a L_2 regularizer to avoid overfitting.

To create the vector representations for each of the word embeddings models considered in this study, we concatenate the hidden state vectors of all the layers for each tagged synset. For both CE and GLM-based models, each gloss was used as a context to build specific contextual word embeddings. If the gloss has more than one tagged token, we take only the first of them for the analysis.

A.2 WordNet metrics: distance

Lets say that we name *"Case-1"* if y is ancestor of x, and *"Case-2"* otherwise. Let $d_W(x, y)$ be the Wordnet distance between two synsets x, y, defined by:

$$d_W(x, y) = \begin{cases} d_{\text{path}}(x, y) & \textit{Case-1}, \\ d_{\text{path}}(x, z) + d_{\text{path}}(y, z) & \textit{Case-2}, \end{cases} \tag{2}$$

where $d_{\text{path}}(x, y)$ is the length of the shortest path between x and y in WordNet, measured in number of hops, and z is the closest common ancestor of x and y in the case that y is not an ancestor of x.

A.3 Minimum-Spanning-Arborescence optimization problem

Given a graph G with nodes N and unknown edges E, we define an auxiliary graph G' with nodes N and edges E', comprised of all possible directed edges. For each edge $e \in E'$, we obtain a prediction h_e that estimates the probability of that

Figure 5: Each triplet is used to create related and unrelated pairs of words according to the relationship *hypernym of*. We create six edge probing pairs, and therefore, the edge probing classifier will need to identify the pair's components and the order in which the words were declared in the pair.

edge representing a valid hypernymy relation, and a distance d_e that estimates the "parent closeness"[3] between the nodes in G.

We define $\delta(v)$ to be the set of edges $\{\langle u, v \rangle : u \in N, u \neq v\}$ where edge $\langle u, v \rangle$ represents a $\langle$parent, child$\rangle$ relation. We also define $\gamma(S)$ to be the set of edges $\{\langle u, v \rangle \in E' : u \notin S, v \in S\}$. We estimate the graph topology of G defined by $E \subset E'$ by solving the following optimization problem:

$$\max_{r \in N} \sum_{e \in E'} x_e h_e \quad \text{s.t.} \quad x_e \in X^* \quad (3)$$

$$X^* = \arg\min \sum_{e \in E'} x_e d_e \quad (4)$$

$$\text{s.t.} \begin{cases} x_e \in \{0,1\} & e \in E' \\ \sum_{e \in \delta(v)} x_e = 1 & \forall v \in N \setminus \{r\} \\ \sum_{e \in \gamma(S)} x_e \geq 1 & \forall S \subset N \setminus \{r\} \end{cases} \quad (5)$$

Objective function (3) is used to find the best root node r; and the nested optimization problem (5) is the minimum spanning arborescence problem applied to the dense graph G'. The final binary values of x_e estimate E by indicating if every possible edge e exist in the graph or not. To solve this optimization problem, we need estimates of h_e and d_e for each edge e. We use the output of the probing classifier as an estimate of the probability of h_e, and use TIM and MCM scores as estimates for d_e (See Section 2.4).

[3]The value of this distance will be small if the hypernym relation is close, or large if it is distant or not valid.

B   Pre-Training corpus comparison

Family	Model	Corpus Size	
		Tokens	Size
NCE	Word2Vec	33B	150GB*
	GloVe-42B	42B	175GB*
GLM	GPT-2	10B*	40GB
	T5	180B*	750GB
CE	ELMo	0.8B	4GB*
	BERT	3.9B	16GB
	RoBERTa	38.7B*	160GB
	XLNet	32.9B	140GB*
	ALBERT	3.9B	16GB

Table 5: Pre-Training corpus sizes used for each one of the studied models. The official sources report corpus sizes in terms of number of tokens or uncompressed size in GB. The symbol * denotes values estimated by us based on official available information. Sizes represents uncompressed corpus sizes.

C   Additional Reconstructed Graphs[4]

Figure 6: Ground Truth Knowledge Graph

[4]Due to space restrictions, the graphs corresponding to Word2Vec, ELMo, T5, BERT will only be included in an extended version of this paper, uploaded to ArXiv
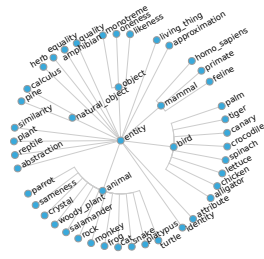
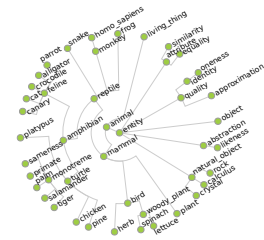Figure 7: GloVe-42B reconstruction using TIM

Figure 8: GPT-2-XL reconstruction using TIM

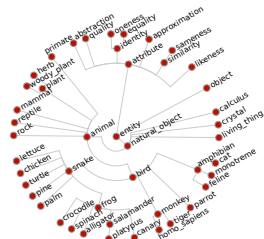Figure 9: RoBERTa-large reconstruction using TIM

Figure 10: XLNet-large reconstruction using TIM

Figure 11: ALBERT-large reconstruction using TIM

D Further information about the impact of semantic factors.[5]

Relative depth in the WordNet graph: (Figure 3-a). For each synset, we compared F1 with depth score (0 % for the root and 100 % for leaves) measuring differences between higher/lower level concepts.

Concept frequency: In Figure 3-c we evaluate if frequent concepts are easier or harder to capture for these models. The frequency was computed by counting occurrences in the 38 GB of OpenWebText Corpus (http://Skylion007.github.io/OpenWebTextCorpus).

Number of Senses and Sense Ranking: (Figure 3-d-e) We studied if models are impacted by multi-sense concepts such as "period", and by their sense ranking (how frequent or rare those senses are). Surprisingly contextualized models, and specially CE models have no significant impact by this factor, suggesting that these models are very effective at deducing the correct sense based on their context. These charts also suggest that these models may be considering context even more than the words themselves. This is intuitive for Masked-Language-Models such as BERT, but not for others, such as GPT-2. Non-contextualized models are impacted by this factor, as expected.
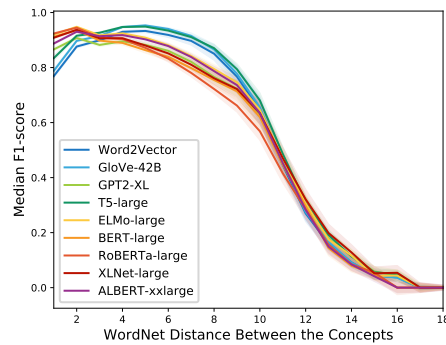

Figure 12: **Graph distance between concepts**: We measured the impact of the number of "hops" that separate two tested concepts on pair-wise F1 score. This chart reveals a strong correlation of all the models in this aspect. As an example of this phenomenon, closer relations such as ⟨chihuahua, dog⟩ are, in general, considerably easier to capture than distant relations such as ⟨chihuahua, entity⟩. For details on how we implement the distance in WordNet, check Appendix A.2.

---

[5]Due to space restrictions, other factors and graphs will only be included in an extended version of this paper, uploaded to ArXiv

E F1-scores of additional categories

Category	W2V	GloVe	GPT-2	T5	ELMo	BERT	RoBERTa	XLNet	ALBERT
abstraction	.7142 ± .1277	.7296 ± .1194	.7224 ± .1897	.7662 ± .0942	.7808 ± .1203	.7718 ± .1582	.7759 ± .1537	.7712 ± .1404	.7635 ± .1732
attribute	.7044 ± .1310	.7213 ± .1237	.7730 ± .0911	.7649 ± .0863	.8093 ± .0886	.8094 ± .0998	.8167 ± .0926	.8064 ± .0891	.8050 ± .0998
communication	.6974 ± .1330	.7251 ± .1224	.7826 ± .0967	.7587 ± .1083	.8049 ± .0925	.8246 ± .0979	.8249 ± .0983	.8066 ± .0987	.8093 ± .1023
group	.7068 ± .1320	.6929 ± .1339	.4972 ± .2955	.7262 ± .1179	.6821 ± .1711	.6173 ± .2399	.6256 ± .2305	.6491 ± .2139	.5858 ± .2745
social group	.7497 ± .1058	.7579 ± .1046	.8155 ± .0883	.7868 ± .0867	.8312 ± .0707	.8516 ± .0742	.8552 ± .0698	.8422 ± .0724	.8556 ± .0819
taxonomic group	.6920 ± .1306	.6648 ± .1330	.3030 ± .2025	.6944 ± .1208	.6011 ± .1583	.4804 ± .2025	.4921 ± .1903	.5371 ± .1944	.4277 ± .2305
family	.7412 ± .1363	.7213 ± .1244	.3131 ± .2003	.6691 ± .1276	.5461 ± .1630	.5626 ± .1537	.5379 ± .1502	.5733 ± .1626	.5437 ± .1705
genus	.6267 ± .0989	.6040 ± .1127	.2567 ± .1582	.7156 ± .1001	.6167 ± .1301	.3696 ± .1857	.4201 ± .1855	.4555 ± .1853	.2862 ± .1945
psychological feature	.7256 ± .1122	.7478 ± .1016	.7829 ± .0904	.7795 ± .0778	.8163 ± .0851	.8181 ± .0954	.8229 ± .0930	.8077 ± .0931	.8208 ± .0915
relation	.7264 ± .1304	.7567 ± .1042	.7612 ± .0809	.7963 ± .0688	.7679 ± .0878	.7662 ± .0995	.7649 ± .0982	.7659 ± .0908	.7727 ± .0929
artifact	.7120 ± .1194	.7389 ± .1068	.7903 ± .0736	.7868 ± .0676	.8308 ± .0693	.8249 ± .0742	.8315 ± .0700	.8319 ± .0702	.8231 ± .0761
covering	.7230 ± .1097	.7510 ± .0970	.7903 ± .0713	.7878 ± .0606	.8398 ± .0599	.8392 ± .0706	.8363 ± .0571	.8393 ± .0576	.8322 ± .0621
instrumentality	.7064 ± .1219	.7378 ± .1052	.7930 ± .0728	.7902 ± .0648	.8308 ± .0676	.8134 ± .0748	.8337 ± .0691	.8313 ± .0711	.8233 ± .0763
device	.7097 ± .1200	.7435 ± .1009	.7956 ± .0713	.7899 ± .0667	.8311 ± .0689	.8198 ± .0701	.8358 ± .0640	.8326 ± .0675	.8230 ± .0743
causal agent	.7240 ± .1137	.7398 ± .1101	.7253 ± .1105	.7751 ± .0757	.8022 ± .0884	.7453 ± .1093	.7631 ± .1056	.7788 ± .1035	.7826 ± .0934
person	.7208 ± .1135	.7351 ± .1113	.7207 ± .1132	.7735 ± .0746	.8022 ± .0854	.7379 ± .1101	.7585 ± .1054	.7779 ± .1053	.7826 ± .0937
living thing	.7295 ± .1112	.7421 ± .1078	.7300 ± .1004	.7862 ± .0749	.8187 ± .0850	.7593 ± .0997	.7823 ± .0922	.7907 ± .0928	.7758 ± .0909
animal	.7349 ± .1049	.7391 ± .1028	.7515 ± .0829	.7837 ± .0714	.8389 ± .0785	.7914 ± .0801	.8179 ± .0701	.8135 ± .0737	.7781 ± .0857
plant	.7445 ± .1044	.7608 ± .0989	.7288 ± .0842	.8168 ± .0653	.8404 ± .0692	.7679 ± .0830	.7962 ± .0688	.7986 ± .0746	.7645 ± .0861
matter	.7402 ± .1117	.7633 ± .1009	.7582 ± .0834	.8002 ± .0662	.7756 ± .0886	.7645 ± .0959	.7614 ± .0941	.7636 ± .0908	.7685 ± .0938
part	.7532 ± .1000	.7759 ± .0852	.7540 ± .0772	.8051 ± .0595	.7580 ± .0844	.7499 ± .0977	.7441 ± .0947	.7526 ± .0880	.7610 ± .0907
substance	.7560 ± .0967	.7791 ± .0809	.7508 ± .0755	.8073 ± .0567	.7542 ± .0828	.7436 ± .0952	.7370 ± .0919	.7477 ± .0862	.7580 ± .0889

Table 6: Each value represents the mean F1-score and standard deviation of all the concepts that belong to each analyzed category. Only the larger version of each model is reported. This is not an extensive list and categories are somewhat imbalanced. Categories were selected based on the number of sub-categories they contained.