Fluorescing[*] Questions: Effects of Semantic Perturbations on BERT Performance in SQuAD 1.1

Spencer Ng, **Lucy Teaford**, **Andy Yang**, and **Isaiah Zwick-Schachter**
{spencerng,lteaford,andyjyang,isaiahzs}@uchicago.edu
CMSC 25610: Computational Linguistics
University of Chicago

1 Introduction

It has long been an outstanding question whether language models have an internal "understanding" of human language or if they are performing some kind of clever pattern matching to make linguistic predictions. In particular, we focus on the degree to which language models rely on the semantic meaning of verbs. For example, if we query a language model for the answer to the question "What year was Beyoncé born in?", will it produce the result "1981" because it found "1981" in conjuction with the verb "born," or simply because "1981" is a year and it made a guess? In particular, if we query the question "What year was Beyoncé photocopied in?", how will that affect the language model prediction?

We propose to evaluate a question answering model with adversarially generated semantic perturbations on verbs in the test sets, in order to see the degree to which the model's success relies on the semantic meanings of verbs which it encounters. The intensity of semantic perturbation will be incremented in a controlled manner, in order to evaluate the degree to which model behavior is dependent on the semantic meaning of verbs.

More specifically, in this paper we consider whether Bidirectional Encoder Representations from Transformers (BERT), a language representation model from researchers at Google AI Language, relies significantly on verb semantics in order to answer questions in the Stanford Question Answering Dataset (SQuAD 1.1). Taking inspiration from Gulordava et al. (2018) and Jia and Liang (2017), we evaluate BERT using adversarially generated test examples of SQuAD 1.1 to determine the properties of its prediction procedure. We train BERT on SQuAD 1.1 to establish a baseline for its performance in answering questions. Then, we generate a set of semantically perturbed but syntactically unchanged test examples by swapping in verbs at differing levels of cosine similarity to the original. This is achieved using a pre-trained word vector model. We evaluate BERT's performance in answering these new semantically perturbed questions to determine whether verb meaning contributes to BERT's understanding of the task. We hypothesize that BERT's performance would change significantly, decreasing in performance as verbs in questions are replaced with increasingly dissimilar verbs, indicating that BERT relies on semantics beyond sentence structure to answer questions.

2 Related Work

Our project is jointly inspired by Gulordava et al. (2018) and Jia and Liang (2017).

2.1 Gulordava et al.

Like Gulordava et al., we are interested in evaluating what a language model learns about language during training. By testing a recurrent long short term memory (LSTM) language model on sensitivity to subject-verb and long-distance number agreement in semantically nonsensical sentences, Gulordava et al. were able to conclude that training on a language acquisition objective is enough to trigger syntactic learning in a recurrent LSTM language model without built-in linguistic knowledge. In our project, we tested for semantic learning in a bidirectional transformer model.

2.2 Jia and Liang

While Gulordava et al. replaced each content word with a random, morphologically equivalent content word to create nonce versions of each original input sentence, we were inspired by Jia and Liang to narrow our focus to explicitly adversarial

[*] Cosine Similarity of Fluoresce and Answer: 0.89

verbs. Jia and Liang, who also evaluate model performance on SQuAD, emphasize the importance of a precise adversarial evaluation scheme in determining what a model truly understands about language.

2.3 Syntax and Semantics: Are they distinct?

We recognize that syntax and semantics are, to an extent, interdependent, and it is difficult to cleanly separate the syntactic and semantic representations of a question. De Bot et al. (2007) assert that language is a dynamic system with complete interconnectedness between its subsystems, which include syntax and semantics. Also calling on dynamic systems theory, Tabor (2021) claims that the syntactic-semantic space is the only level of representation in language, and that syntactic and semantic language processing cannot be distinguished. Shen et al. (2017) propose a novel neural language model, Parsing-Reading-Predict Networks (PRPN), that learns the syntactic structure of sentences, then uses this structural information to form better semantic, or word meaning, representations. The proposed mechanism underlying this model suggests an interconnectedness between syntax and semantics.

However, we are encouraged by literature that suggests a distinction between syntax and semantics. Ek et al. (2019) trained LSTM language models on sentences annotated with syntactic roles and semantic information, and they found that only the syntactic tags lowered the perplexity of the models. Huang et al. (2021) present ParaBART, a sentence embedding model that learns to disentangle syntax and semantics in input sentences through paraphrasing. Mitchell and Steedman (2015) found that distributional models produce word representations that are decomposed into orthogonal syntactic and semantic spaces.

Given these findings, we are confident that language representation models are capable of differentiating syntactic and semantic information during training, and that these subsets of information have distinguishable effects on the success of these models in completing natural language understanding tasks.

3 Methods

3.1 Task

The SQuAD 1.1 dataset is designed to evaluate reading comprehension, consisting of a series of "context" paragraphs drawn from 536 English Wikipedia articles and a series of questions whose answers can be found as a segment of text from the corresponding article (Rajpurkar et al., 2016). Questions and their related paragraph are provided as inputs to natural language processing models, which then extract the answer from the provided paragraph. We use the publically-available SQuAD 1.1 dataset for our experiments, training on 19,047 paragraphs of context and their associated 87,599 questions from the training set. For evaluating our model and creating perturbed questions and contexts, we used the dev set, consisting of 10,570 questions and 2,137 related paragraphs.

3.2 Model

We trained and evaluated a baseline model fine-tuned from a pretrained BERT uncased language model and tokenizer using the Huggingface Transformers library (Wolf et al., 2020). The bidirectional BERT model is used because it is well-suited to accurately model downstream tasks, and the parameter fine-tuning process is relatively inexpensive (Devlin et al., 2019). The BERT model was trained for three epochs on the SQuAD 1.1 training set, using a learning rate of $2 \cdot 10^{-5}$, a batch size of 4, and a weight decay of 0.01. Our baseline model achieves an accuracy of 80.04%.

Example contexts and questions were broken into tokens (text spans) of at most 384 characters, and the start and end indices of ground truth answers within paragraph contexts were fed into the model. For generating predictions and evaluation, a similar process was followed, where tokens were generated from questions and contexts, and the model output a vector of probabilities S and E corresponding with each character in the tokenized context. Each probability S_i and E_j then corresponded with the chance of the ith character in the context span being the first character of the answer to the question and the jth character being the last character of the answer. We apply a softmax function to each vector to find the most likely start and end locations for the answer, ensuring that $i < j$, and outputting the character sequence with the ith through jth characters from the span as the answer. If no valid prediction is found, we output an empty string as the answer.

3.3 Semantic Perturbation

To control for a model's sensitivity to syntactic structure while testing its sensitivity to semantic perturbation, we used the following procedure to modify the meaning of a piece of text while preserving its syntactic structure:

1. Select a verb v from a piece of text
2. Lemmatize v.
3. Use a word-vector model to find a second verb v' at a desired cosine similarity distance from the lemmatized v
4. replace v in the text with v'

In this way, we can ensure that the syntax of the text remains constant while the semantic meaning changes. We use the NLTK library (Bird et al., 2009) for tokenizing sentences, tagging words with their POS, and lemmatizing verbs. Verb tenses that the NLTK tagging library could recognize included were past tense, present participle, past participle, non-3rd person singular present, and 3rd person singular present.

We use the gensim library (Rehurek and Sojka, 2011) for computing verbs of desired cosine similarity values. We imported the WebVectors enwiki_upos_skipgram_300_3_2019 model (Kutuzov et al., 2017) to use as our word vector model. This is a Gensim Continuous Skipgram model, trained on the English Wikipedia dump of October 2019. It contains 249212 different words, tagged by part of speech. The choice of a vector model trained on Wikipedia articles is to keep our verb transformations within the same register, as SQuAD1.1 is also from Wikipedia, Given a verb v, we find other verbs v' of varying cosine similarities to v by iterating through the verbs in the model.

To conjugate verbs into the desired tense we use the mlconjug library (Diao, 2021). This allows us to take a verb v' of the desired cosine similarity to our original verb, give it the same syntactic category as the verb it is to be replacing, and then insert it into the text.

Through manual inspection we verified that this process was successful at creating sentences that were semantically different but syntactically similar to the original. In particular, as the cosine similarity of the transformed verb decreases, the inspected semantic meaning becomes more and more anomalous. An example of this is placed in Table 1.

Cosine Similarity	Question
1.0	What percentage of Australia's dairy cattle are **found** in Victoria?
0.6	What percentage of Australia's dairy cattle are **identified** in Victoria?
0.5	What percentage of Australia's dairy cattle are **searched** in Victoria?
0.4	What percentage of Australia's dairy cattle are **disproved** in Victoria?
0.3	What percentage of Australia's dairy cattle are **uncoound** in Victoria?
0.2	What percentage of Australia's dairy cattle are **fluoresce/outbidden** in Victoria?

Table 1: Question id = 3006 modified at varying levels of verb cosine similarity (1.0 indicates the original question in the SQuAD1.1 test set).

3.4 Data Preparation

To prepare the data for our experiments, we needed to process it into a format that was more easily consumable by our models and our verb substitution process. To do this, we first extracted the training set to feed it to our models, and then we extracted and processed the test (dev) set to use to evaluate out models. We extracted question answer pairs for easier verb substitution and we processed each question answer grouping into a category. All questions in the SQuAD 1.1 dataset belong to one of 10 categories: "date," "other numeric," "person," "location," "other entity," "common noun phrase," "adjective phrase," "verb phrase," "clause," or "other."

We could not locate a dataset that tagged the questions for us, so we categorized them ourselves. We did this by examining questions in the test/dev dataset by hand and guessing at patterns that could be easily checked to determine a questions category. There were several patterns that appeared to be reliable indicators of the questions category, for example questions containing "how many" will typically have "numeric" answers and questions asking about a date/year/month will typically be referring to a date. Based on our manual examination of questions and extending that using regular expression patterns, we classified approximately 1800 questions, accounting for nearly 20% of the questions in the test/dev set. Getting a measure of our accuracy would be challenging, since these were already hand classified, but we estimate about 75% accuracy on the questions we could determine a category for. The other 8000 that we could not classify were categorized as "other".

3.5 Evaluation

To evaluate the accuracy of our model on both the baseline test set and perturbed examples, we compute accuracy and F1 score metrics.

Given our fine-tuned model f and a set of test questions Q, related contexts C, and ground truth answers $\mathcal{A}$, we define $\hat{A}$ to be the set of answers predicted by f, where $\hat{a}_i = f(q_i, c_i)$ is the predicted answer to the ith question in Q, which has an associated context c_i. We then apply a normalizing function n to each $\hat{a}_i \in \hat{A}$, where we remove articles, punctuation, and white space from answers. We also ensure each predicted answer and associated set of ground truth answers $A_i \in \mathcal{A}$ is converted to lowercase to account for the original uncased BERT model. This normalizing function n is borrowed from the official evaluation script from Rajpurkar et al. (2016).

The *accuracy* as reported in our results for a given test set $T = (Q, C, A)$ is then equal to

$$\text{acc}(T) = \frac{1}{|Q|} \sum_{q_i \in Q} \begin{cases} 1 & \text{if } f(q_i, c_i) \in A_i \\ 0 & \text{otherwise} \end{cases}$$

Similarly, the F1 score is equal to

$$\text{F}_1(T) = \frac{1}{|Q|} \sum_{q_i \in Q} \text{F}_{1_a}(f(q_i, c_i), A_i),$$

where F_{1_a} is the F1 function provided by Rajpurkar et al. (2016) that measures the m.aximum overlap between a given prediction predicted answer $\hat{a}_i$ and all ground truth answers $a \in A_i$.

To compute the accuracy and F1 score for a given category of questions in our analysis, we take a subset of the test set questions and associated contexts $Q' \subset Q$ that belong in that category.

4 Experiments

After training BERT on the SQuAD 1.1 question and answer task, we generate adversarial examples to evaluate the model upon. The original SQuAD 1.1 test set contains 10570 questions, each referring to one of 2067 paragraphs. Three classes of test sets were generated from the original: one modifying only the questions, one modifying only the paragraphs, and one modifying both. Within each class, we generated 5 different gradations of semantic perturbation. We run BERT on these test sets in order to evaluate the effect of different kinds and degrees of verb perturbation.

4.1 Generating Semantically Perturbed Examples

The generated semantic perturbations followed the process outlined in the methods section in the manner described below:

Questions: The last verb in each given question was perturbed.

Paragraphs: For every question which refers to a given paragraph, we find occurrences of the last verb in the question within the paragraph. Each of these occurrences is replaced with the verb which replaces the question's verb. We insure that each replacement verb in the paragraph is in the same tense as the original verb.

Original	**Perturbed**
Warsaw **is** located on two main geomorphologic formations ...which **divides** the city into two parts, left and right. The left one **is** situated both on the moraine plateau ...**called** Warsaw Escarpment. It **is** 20 to 25 m (65.6 to 82.0 ft) high in the Old Town ...	Warsaw **needs** located on two main geomorphologic formations ...which **edomiinds** the city into two parts, left and right. The left one **needs** situated both on the moraine plateau ...**tound** Warsaw Escarpment. It **needs** 20 to 25 m (65.6 to 82.0 ft) high in the Old Town...

Table 2: Excerpt of paragraph id = 87 modified with a verb cosine similarity of 0.4, replacing the the verbs which occur in the questions referring to said paragraph.

In this manner, the modified questions and modified paragraphs are intended to be consistent with each other.

Within the English Wikipedia 2019 word vector model, the cosine similarity between any given verbs generally ranges between 0.2 and 0.6. For this reason, we generated 5 different semantically perturbed test sets, substituting in verbs at the

Cos. Similarity	**Baseline**		**0.2**		**0.3**		**0.4**		**0.5**		**0.6**	
Result	F1	acc	F1	acc	F1	acc	F1	acc	F1	acc	F1	acc
Question Only	87.79	80.04	83.48	75.01	81.81	73.46	82.51	74.20	85.76	77.73	86.14	78.25
Context Only	87.79	80.04	84.36	74.27	85.40	76.36	85.66	76.92	86.27	77.69	86.61	78.16
Question + Context	87.79	80.04	81.81	72.04	80.96	71.94	81.98	72.92	85.18	76.59	85.88	77.44

Figure 1: F1 and accuracy results for increasing levels of cosine similarity when only the question is modified, only the context is modified, or both the question and context are modified.

following different levels of cosine simililarity: $0.6, 0.5, 0.4, 0.3, 0.2$. Table 1 illustrates a semantically perturbed question, Table 2 illustrates an example of a semantically perturbed paragraph.

5 Results

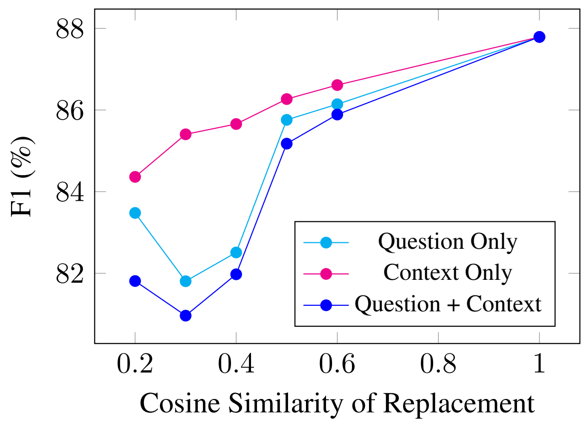

Figure 2: F1 (%) for increasing levels of cosine similarity when only the question is modified, only the context is modified, or both the question and context are modified.

In our experiments, looking at the relationship between cosine similarity and accuracy, we saw very similar results to what we hypothesized.

5.1 Overall Results

For cosine similarity levels 0.3 through 0.6 we saw an approximately linear decreasing relationship between the cosine similarity and the accuracy of the model. This decreasing trend shows that our model does utilize semantic meaning when predicting answers to the SQuAD1.1 question tasks. In our experiments, the prediction accuracy ranged from a low of approximately 75% accuracy to a high of nearly 90% accuracy. Because the accuracy did not dip below 75%, we are led to believe that there is still a fairly strong reliance on syntactic structure, otherwise we would expect the drop to be much larger.

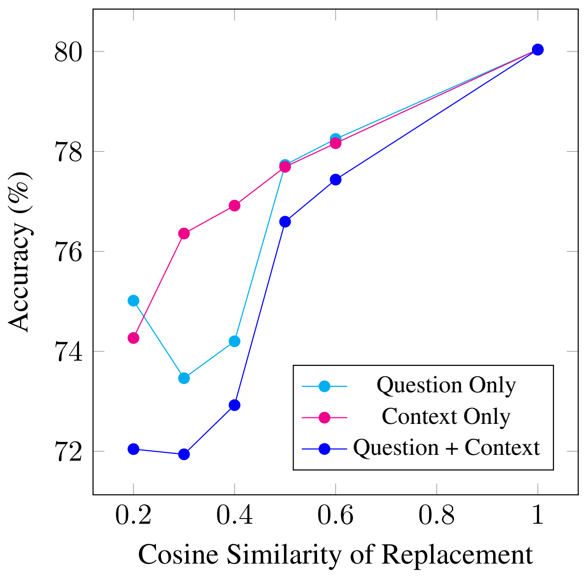

Figure 3: Accuracy for increasing levels of cosine similarity when only the question is modified, only the context is modified, or both the question and context are modified.

However, one outlier we noticed was at the 0.2 cosine similarity level. When modifying the question (with or without modifying the context) we saw a sharp increase in accuracy where before there had been a decrease. We saw the sharpest change when looking at modifications to the question only. The trend remained linearly downward for the context only category, and the question + context category had a less sharp increase. This leads us to believe that this increase is due to the change in the question. We are unsure why the question changes at a 0.2 cosine similarity level would cause this when none of the higher cosine similarities did.

Cos. Similarity	**Baseline**		**0.2**		**0.3**		**0.4**		**0.5**		**0.6**	
Result	F1	acc	F1	acc	F1	acc	F1	acc	F1	acc	F1	acc
common noun phrase	85.70	77.04	81.69	73.90	81.19	73.27	83.34	74.84	83.82	75.16	84.60	75.47
date	95.94	95.65	96.09	96.09	96.09	96.09	96.09	96.09	95.94	95.65	95.51	95.22
other	87.19	78.88	82.41	73.19	80.75	71.72	81.69	72.77	84.90	76.24	85.27	76.81
other numeric	90.58	84.86	89.86	84.62	89.77	84.16	89.94	84.16	90.36	84.86	90.88	85.20
person	91.98	88.67	87.57	83.15	78.94	74.59	74.83	69.89	90.10	86.46	90.86	87.29

Figure 4: F1 and accuracy results for increasing levels of cosine similarity across five question categories.

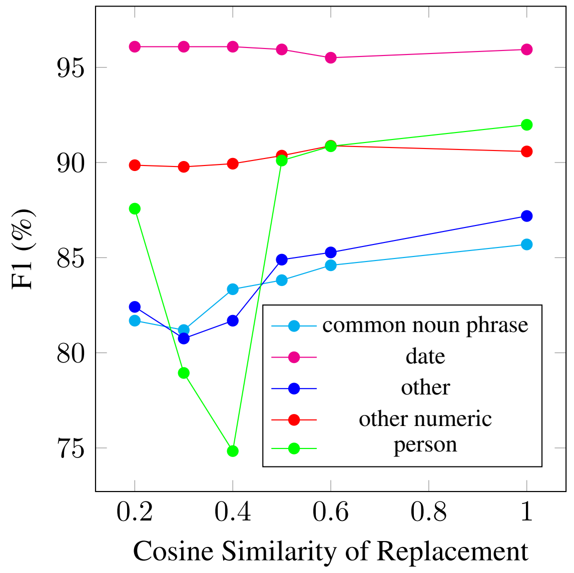

Figure 5: F1 for increasing levels of cosine similarity across five question categories.

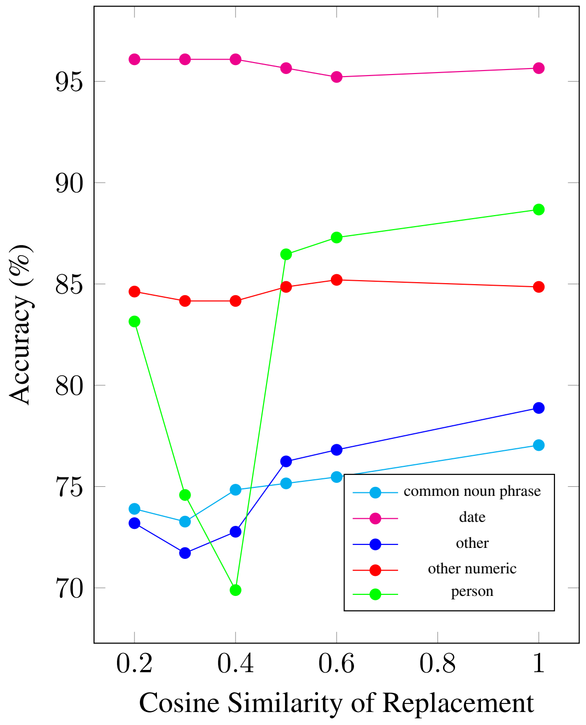

Figure 6: Accuracy for increasing levels of cosine similarity across five question categories.

5.2 Question Category Analysis

When looking at these trends in five of the ten categories discussed earlier ("common noun phrase," "date," "other," "other numeric," and "person"), we saw that some of these categories of questions were impacted much more severely by our substitutions, while others remained relatively constant. The "other" category is the largest of the categories and that line serves largely as a reference for the general trends in the dataset, as it contains all of the questions we could not categorize through data processing.

The categories we saw the least change in were "date" and "other numeric", which makes sense as for this sort of question we'd expect the model to lean more heavily on syntax. We believe that answers to those kinds of questions are fairly obvious from a context's syntax alone. The "common noun phrase" category seemed to largely mirror "other" (the general trends in the dataset), which makes sense due to the wide range of questions that have answers that are "common noun phrases". The category we saw the steepest change in was "person." This is the most interesting result from our category analysis, as these questions were fairly clear cut (all containing some variation of "who," "who is," or "whose"), and this result is very unexpected. It seems unperturbed at 0.6 and 0.5, and then falls by 15 percentage points at 0.4 cosine similarity, before beginning to climb back up as the cosine similarity drops further.

This category analysis shows that different types of questions are affected by perturbations in the semantic meaning in different ways. For example, date and numeric based questions are not affected at all, while questions about people are strongly affected.

6 Discussion

We drew two important conclusions from our experiments. First, while we saw a general trend in decreasing accuracy with decreasing cosine similarity, the model still performed close to baseline performance. This suggests that verb semantics is not essential to the model's prediction mechanism, as preservation of syntax is enough to keep the results within a 10% range of baseline performance.

Second, we found that question type had a significant effect on the model's performance on adversarial inputs. We saw a significant decrease in accuracy in answering perturbed "person" questions, and an overall increase in accuracy when answering perturbed "date" questions. Clearly, the relative importance of syntax and semantics to model performance on language understanding tasks depends on the nature of the task, and even the specific conformation of the inputs within one task.

7 Conclusion and Future Work

Our analysis left us with some questions that we would be interested in exploring in future courses. The first major question relates to the upward spike we saw at 0.2 cosine similarity in Figures 1 and 2. We assume that this is due to the mid-range of cosine similarities being the most chaotic, while the lower cosine similarities start to look more like similar-but-opposite words like antonyms. However, we aren't able to confirm this without a more complete analysis of how our vector space behaves and what our cosine similarity selection is actually doing in terms of meaning.

The second major question is based on our category analysis. We saw very different behaviors dependent on the category of the questions. This begs the question of why some questions are more impacted than others. We proposed some possible explanations for this in our results section, but further work would be required to fully understand why these categories behave the way they do under the transformations we've applied. Specifically, we believe it would be useful to further divide the categories into smaller subcategories to narrow down which questions are and aren't impacted and examine the causes.

Our process for categorizing questions could also be improved upon and a larger group of questions could be categorized. We were only able to categorize less than 20% of the dataset, and so the category analysis we were able to do was limited. More sophisticated and complete categorization could lend deeper insight into the behaviors we're seeing based on our perturbations.

References

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit.* " O'Reilly Media, Inc.".

Kees De Bot, Wander Lowie, and Marjolijn Verspoor. 2007. A dynamic systems theory approach to second language acquisition.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Sekou Diao. 2021. mlconjug3. *GitHub. Note: https://github.com/SekouDiaoNlp/mlconjug3*.

Adam Ek, Jean-Philippe Bernardy, and Shalom Lappin. 2019. Language modeling with syntactic and semantic representation for sentence acceptability predictions.

Kristina Gulordava, Piotr Bojanowski, Edouard Grave, Tal Linzen, and Marco Baroni. 2018. Colorless green recurrent networks dream hierarchically.

James Y. Huang, Kuan-Hao Huang, and Kai-Wei Chang. 2021. Disentangling semantics and syntax in sentence embeddings with pre-trained language models.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems.

Andrei Kutuzov, Murhaf Fares, Stephan Oepen, and Erik Velldal. 2017. Word vectors, reuse, and replicability: Towards a community repository of large-text resources. In *Proceedings of the 58th Conference on Simulation and Modelling*, pages 271–276. Linköping University Electronic Press.

Jeff Mitchell and Mark Steedman. 2015. Orthogonality of syntax and semantics within distributional spaces.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text.

Radim Rehurek and Petr Sojka. 2011. Gensim–python framework for vector space modelling. *NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic*, 3(2).

Yikang Shen, Zhouhan Lin, Chin-Wei Huang, and Aaron C. Courville. 2017. Neural language modeling by jointly learning syntax and lexicon.

Whitney Tabor. 2021. On the relationship between syntactic and semantic encoding in metric space language models.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.