Team Mxgra at SemEval-2020 Task 4: common sense making with next token prediction

Kris Collins
kristopher.collins@student.uni-tuebingen.de

Max Grathwohl
max-peter.grathwohl@student.uni-tuebingen.de

Heba Ahmed
hebah.ahmed@student.uni-tuebingen.de

Abstract

In this paper, we explore solutions to a common sense making task in which a model must discern which of two sentences is against common sense. We used a pre-trained language model which we used to calculate perplexity scores for input to discern which sentence contained an unlikely sequence of tokens. Other approaches we tested were word vector distances, which were used to find semantic outliers within a sentence, and siamese network. By using the pre-trained language model to calculate perplexity scores based on the sequence of tokens in input sentences, we achieved an accuracy of 75 percent.

1 Introduction

Much research has been conducted on Natural Language Understanding (NLU) and models based on neural networks have achieved state-of-the-art results in reading comprehension . However, results in tasks regarding common sense making do not rival those of human performance, such as in inference tasks in which state-of-the-art models score upwards of 80 percent (Devlin et al., 2019).

As the organizers of the SemEval 2020 Commonsense Validation and Explanation (Wang et al., 2020) task highlighted, pinpointing exactly why an utterance is against common sense is not a trivial undertaking. Advances in this area could lead to improvements in NLU.

We participated only in Task A, which was determining which of two sentences was against common sense. For example, given the input "He poured milk on his cereal" and "He poured orange juice on his cereal" we would identify "He poured orange juice on his cereal" as the sentence against common sense. Task A was evaluated by accuracy. All data was in English.

We used a pre-trained language model to calculate the probability of the next word in a sequence to produce perplexity scores for each of the sentences in the input pairs. A higher perplexity score meant a lower overall probability of the words occurring in this order, thus being more likely against commonsense.

We also attempted to solve this task with word vector distances, attempting to find an outlier among tokens in each input sentence, and with a siamese network, which was trained as a feature embedder for valid sentences.

2 Background

The data consisted of pairs of English sentences, one of which made sense and the other did not. The desired output was the index of the sentence against common sense. For example, the model would take the input sentences "He poured orange juice on his cereal." and "He poured milk on his cereal." and output a 0. The official training data from the ComEV organizers was a collection of 10,000 pairs of sentences.

Proceedings of the 14th International Workshop on Semantic Evaluation, pages 569–573
Barcelona, Spain (Online), December 12, 2020.

2.1 Pre-Trained Models

Though there are applications that intend to achieve machine sense-making via designing conceptual knowledge bases (Periñán Pascual and Arcas Túnez, 2007; Liu and Singh, 2003), language models pre-trained on large corpora are still widely applied as they demonstrate high scores of accuracy (Wang et al., 2018; Wang et al., 2019).

When the fine-tuned ELMo was applied to a SemEval 2018 common sense dataset, it outperformed previous state-of-the-art performance by scoring 74.1 percent accuracy (Wang et al., 2019). These results inspired us to use the power of language models as the main force behind our model.

3 System Overview

We used three distinct and separate approaches to solve this task. Our main approach is based on the probability of the next word in the sequence. After analyzing the data, we realized the structure of the input was such that the perplexity score of each sentence would be a strong indicator of its validity. We made use of a pre-trained neural network, the OpenAI GPTLM Head Model (Radford et al., 2018b), from the Transformers library from HuggingFace[1] to calculate perplexity scores. This model is based on the OpenAI GPT model (Radford et al., 2018a), a causal (unidirectional) transformer pre-trained using language modeling on a large corpus with long range dependencies, the Toronto Book Corpus (Zhu et al., 2015). The OpenAI GPT Head Model consists of the basic transformer (Vaswani et al., 2017), but with a language modeling head on top (linear layer with weights tied to the input embeddings) and is therefore powerful at predicting the next token in a sequence. Our classification pipeline was rather small and straight-forward. We loaded the model with pretrained weights and fed one tokenized sentence as input. As output we took the prediction scores of the language modeling head (e.g. the scores for each vocabulary token before a SoftMax layer that chooses the most probable word). The scores for both sequences of a pair were then compared, the larger one labeled as invalid.

Another approach was to calculate distances between word vectors of words inside one sentence. The general idea is simple: as language models trained on large corpora are increasingly better at embedding words in semantically meaningful vectors, we tried to use the multidimensional representation of big word embeddings to find semantic outliers in the sentence. If a sentence contained such an outlier, it is marked as the invalid one. Only sentences where the keyword (differentiating word between the two sentences) was a subject or object (direct and indirect) were used. The BERT base-uncased language model (Devlin et al., 2019) and the last four of its layers computed the word embeddings. We used the spaCy parser (Honnibal and Montani, 2017) to first create a dependency tree of the sentence, a simple function then extracted the root of the sentence and the desired subjects and objects. Next, we computed the cosine distance between the word vectors in the sentence and averaged the score over the number of tokens. The smaller this score, the more likely the sentence to be sensical.

Born out of the word-vector approach discussed above, our final approach was a siamese network with which our goal was to train it as a feature embedder for valid sentences. From a selection of the provided data we crafted anchor sentences through the BERT Masked LM model (Devlin et al., 2019). The keyword was masked and of the ten most probable predicted words the most viable (same category of part of speech) were used. This was done automatically for all sentences, therefore there is the possibility that some novel sentences are not viable as sensible anchors. A review of all the anchors was not feasible. We implemented a custom variant of the triplet loss based on vector distances, which should push the model to create context embeddings that achieve a higher cosine similarity for the valid sentences to the anchor than the invalid ones.

4 Experimental Setup

We limited preprocessing because of the terseness of input data. The shortest input string was of length two. The longest input string was 23 words long. The average length of input was 7.7 words. We decided to alter the data as little as possible in the preprocessing stage because of our next-token-prediction

[1] https://huggingface.co/transformers/

approach. Input without stop words, for example, would have resulted in lower perplexity scores from the GPT Head Model (Radford et al., 2018b), as they would be more unlikely to appear in the corpus the model was trained on. Additionally, some of the input strings in the training set were against common sense because they were grammatically incorrect, and we opted to preserve the original structure of the input so that these sentences would result in a higher perplexity score from the model.

We performed a contraction mapping with the use of the pycontractions package[2] to ensure proper tokenization. Task A was evaluated on accuracy.

5 Results

The GPT model achieved an accuracy of 75 percent, which places us 25th among 28 teams, who submitted results in the post-evaluation phase. Teams placed 15th to 26nd, achieved accuracies between 80 and 90 percent. Teams 14th and higher achieved accuracies above 92 percent, the highest being 96.7 percent.

Model	Accuracy
GTP Head Model	75%
Word Vector Distance	61%
Siamese Network	51%
Random Baseline	50%

Table 1: Various models we used to solve Task A and their accuracies on the development dataset.

5.1 Error Analysis

Though the initial score of 75 percent was encouraging, after an error analysis it is our conclusion that this model alone could not reach a higher accuracy without the use of additional modules or methods. Our model performs a superficial evaluation of the validity of input sentences, measuring how likely each word is to appear after another based on a corpus, and does not delve into semantics, which led to erroneous labeling of sentences that were not against common sense but unusually worded or used infrequently occurring words.

We performed the analysis on randomly selected wrongly predicted sentences which were representative of the various sentence structures across the whole dataset. An error analysis over the entire set of incorrect predictions is beyond the scope of this paper. As a result, we cannot provide exact numbers or percentages of the error frequencies.

Unusual, but not wrong
We noticed a high frequency of input sentences that were not ungrammatical nor nonsensical, but phrasing or peculiar words that lay outside of what could be considered common parlance. For example, "Sewerage is very important for cities," employs the less popular "sewerage" over "sewage." Far more common were oddly phrased inputs, such as "I use swords", "A niece is a person", "I read stars", or "People write with pens." These unusual, though grammatically correct, inputs were difficult for the model because the training data were books, which no doubt included utterances that would be considered common parlance.

Erroneous input
Some of the input sentences were simply ungrammatical. For example, a possessive "'s" is employed instead of a pluralizing "s," there was subject-verb disagreement or incorrect spellings present in the input.

Both Wrong/Right
There were also instances of input in which both sentences could be against common sense or with it. For example, "Sugar is sweet for humans." and "Sugar is bad for humans."

[2]pycontractions v.2.0.1. https://pypi.org/project/pycontractions/

Last word changed
We found these sentences to be particularly tricky for our model. The probability of these sentences are identical until the last word of the sentence, making this essentially a unigram probability problem. For example with the sentence pair, "The baby played with fire" and "The baby played with blocks." the correct label comes down to the probability that "blocks" or "fire" follows "with" because the rest of the sentence would have an identical score up to the point of derivation.

Multiple differences
Unlike the rest of the input sentences, which swapped one word for another or saw the flipping of subject and object, sentences which employed multiple differences across sentence 0 and sentence 1 were difficult for the model to handle.

For example, in "People should wear sunglasses when they are short-sighted" and "People should wear glasses when they are myopic" we see the problematic word "sunglasses" in sentence 0 replaced in sentence 1 and a synonym for "short-sighted" is also used.

While a pattern can be seen in the common erroneous predictions made by our model, the underlying problem is that the analysis of sentences was superficial.

For the word vector distance approach, a couple of factors were observed during our experiments. Obviously the way of finding related words to compare inside the sentence is significant. We believe that with a more sophisticated method to find relevant tokens to compare our keywords to this score could be improved. Problematic for this approach are of course sentences in which only word order is changed to create a nonsensical sentence. We also found that some word vectors measured a near distance that we would intuitively not have guessed. This could be due to the context of learned information from the BERT model, meaning that some particular themes could be overrepresented when applied to the context of a general discourse. Intuitively in the sentences "I put a turkey in the fridge" and "I put an elephant in the fridge", "elephant" and "fridge" should measure a longer distance than their counterparts "turkey" and "fridge". If the model (the word embedder) was trained on large amounts of wildlife data rather than American tradition, it would score "elephant" and "fridge" closer together.

One key problem with the siamese network was the anchor sentences, which require a good amount of effort and time investment if done properly. Another origin of low performance is possibly due to the variety of sentence structure and the very limited data two sentences offer in comparison to document similarities (from which inspiration for this approach was drawn). With many sentences that share a similar structure and length, the model seemed to perform well for a subset of the overall dataset, but lost progress when encountering a novel structure. We theorize therefore that the model couldn't learn to differentiate between the two classes invalid and valid, but rather between different sentence structures.

6 Discussion

The main observation we drew from the results is that our model is not able to handle input sentences that are outside of common parlance. Our model also had difficulty correctly labeling sentences with spelling or grammatical errors i.e "dog's" instead of "dogs."

On the other hand, our model is able to distinguish very well the nonsensical sentence in pairs in which a reasonable noun is exchanged for an outrageous one. For example, "book" was replaced with "tiger" in the sentence "I read a ___".

We came to the conclusion that, if we were able to repeat this task, we would use a hybrid approach. Rather than relying on statistics, however powerful and effective, we would supplement our process with a discriminative function. Our model could also benefit from the utilization of a semantic analysis. With the use of, for example, FrameNet (Baker et al., 1998), a lexical database of annotated English examples of how words are used in actual context from the International Computer Science Institute at UC Berkeley, we could establish that, using the above example, a tiger is an animal, and therefore something that cannot fit in with the semantic framework for reading.

7 Conclusion

By using the GPT Head Model from OpenAI (Radford et al., 2018b) to calculate perplexity scores for input sentence pairs we were able to achieve an accuracy of 75 percent, placing us 28th of 28 teams in the task. Our model had difficulty with nuances and only captured superficial representations of meaning. This system could be improved through the addition of discriminative processing and semantic analysis.

References

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet project. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics - Volume 1*, ACL '98/COLING '98, page 86–90, USA. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, jun.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. *To appear*, 7(1).

Hugo Liu and Push Singh. 2003. OMCSNet: A commonsense inference toolkit. *Submission. Available at: http://web. media. mit. edu/˜ hugo/publications*.

José Carlos Periñán Pascual and Francisco Arcas Túnez. 2007. Cognitive modules of an NLP knowledge base for language understanding. *Procesamiento del Lenguaje Natural*, (39):197–204.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018a. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf*.

Alec Radford, Karthik Narasimhan, Time Salimans, and Ilya Sutskever. 2018b. Improving language understanding with unsupervised learning. *Technical report, OpenAI*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Liang Wang, Meng Sun, Wei Zhao, Kewei Shen, and Jingming Liu. 2018. Yuanfudao at SemEval-2018 task 11: Three-way attention and relational knowledge for commonsense machine comprehension. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 758–762, New Orleans, Louisiana, jun. Association for Computational Linguistics.

Cunxiang Wang, Shuailong Liang, Yue Zhang, Xiaonan Li, and Tian Gao. 2019. Does it make sense? and why? a pilot study for sense making and explanation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4020–4026, Florence, Italy, jul. Association for Computational Linguistics.

Cunxiang Wang, Shuailong Liang, Yili Jin, Yilong Wang, Xiaodan Zhu, and Yue Zhang. 2020. SemEval-2020 task 4: Commonsense validation and explanation. In *Proceedings of The 14th International Workshop on Semantic Evaluation*. Association for Computational Linguistics.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27.