Advances in Cognitive Systems 10 (2022) 1-6 Submitted 9/2022; published 11/2022

Asking Language Models to Explain Pronoun-Ambiguous Sentences

Yida Xin YXIN@BU.EDU
Department of Computer Science, Boston University, Boston, MA 02215 USA

Henry Lieberman LIEBER@MEDIA.MIT.EDU
Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139 USA

Pedro Colón-Hernández PE25171@MIT.EDU
Media Lab, Massachusetts Institute of Technology, Cambridge, MA 02139 USA

Abstract

In recent years, large language models have proven powerful at various tasks and successfully dominated the attention of numerous communities in academia and industry. At the same time, people also worry more about the consequence of having such opaque models increasingly taking over critical, human-centered applications. In this paper, we propose ProGeneXP, a low-cost general-purpose framework where a human user can instruct a language model to learn to generate natural language explanations of intended meanings of natural language inputs. By *explanation*, we mean natural language texts that convey what should be commonsense to even the laypeople. The ability to explain what they see in commonsensical ways, not only renders a form of transparency into the language models themselves, but also is a desirable general property for systems that can be regarded as cognitive and symbiotic with humans. To evaluate how such model-generated explanations help with performance, we test them downstream on a large pronoun disambiguation dataset. Our preliminary result suggests that transparency and performance can be improved together, undermining the credibility of the idea that there is a necessary trade-off between them.

1. Introduction

In this paper, we argue that AI models can leverage both performance and transparency, setting the stage for human-like cognitive systems. Specifically, we propose ProGeneXP, a low-cost general-purpose framework that leverages large language models' ability to draw meaningful summary, to fine-tune them so they learn to generate pertinent descriptions of their input texts.

In ProGeneXP, a human user not only provides the said input texts, but does so in such a way that instructs the language model to achieve desired learning outcomes. The learning process fully takes advantage of the model's implicit representations of language semantics acquired through painstaking pre-training. The user simply instructs, in a Supervised Learning fashion, by providing natural language explanations as input labels paired with natural language utterances as input data. What we mean by *explanation* here is, very simply and loosely, natural language utterances that resemble what people would regard as conveying necessary commonsensical or background

information. As we will see in section 2.2, an example of our explanations is for the sentence "The infection spread throughout the building because it was airborne." To correctly resolve the pronoun "it" to co-refer to the entity "infection" instead of "building," we can provide ProGeneXP with the explanation, "Airborne viruses can spread quickly in ventilated in-door areas."

Albeit primitive, we believe this sheds light on building natural language explanation user interface for the future. Also, the ability to explain what they see in commonsensical ways, not only renders a form of transparency into the models themselves, but also is a desirable general property of systems that can be regarded as cognitive and symbiotic with humans. Our hopes are that the model learns from the initial instructions to generate explanations closer and closer to user-intended meanings, and conversely the user learns to better probe and prime the model.

We design ProGeneXP to specialize in the pronoun disambiguation task and generates task-specific natural language explanations. The two stages of ProGeneXP are shown in Figure 1.

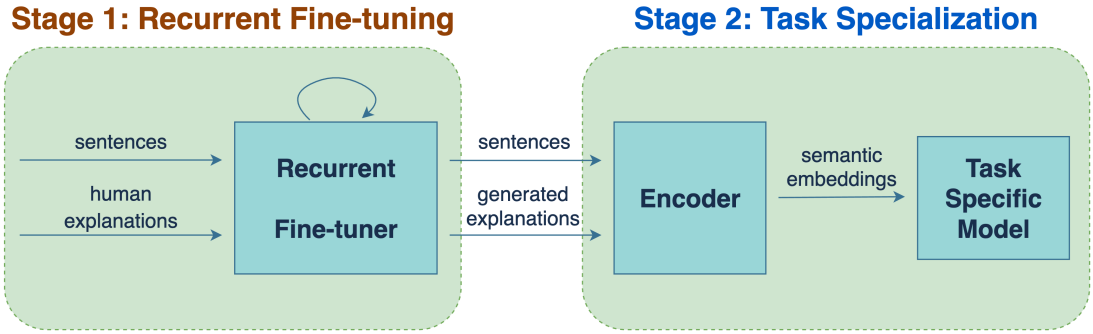

Figure 1: Architecture of ProGeneXP. The self-loop at the top of the Recurrent Fine-tuner indicates the recurrent style learning that we detail in section 2.1.

Both stages are model-agnostic, meaning one can instantiate these stages with any language models of their preference[1]. Overall, ProGeneXP is low-cost for human developers, because of both its straightforward architecture and the suite of publicly available pre-trained language models. ProGeneXP is also low-cost for human users, because users can initiate ProGeneXP's fine-tuning pipeline using only a handful of guiding explanations.

For the remainder of the paper: In section 2, we introduce and explain the two stages of ProGeneXP; the first stage yields a fine-tuned language model that has learned to generate desired explanations, and the second stage evaluates the generated explanations (from the first stage) on a downstream pronoun disambiguation task. In section 3, we discuss additional details of our experiments and our preliminary results. In section 4, finally, we reflect on our approach, mention related work, and postulate future work.

[1]In section 3, admittedly, the reader will see that specific language models are used. We emphasize that the use of specific models for experimental purposes does *not* undermine the model-agnosticism of ProGeneXP.

2. Recurrent Fine-tuning

Recurrent Fine-tuning (RFT) is our attempt to reconcile AI transparency and interpretability, which gives the impression of resorting back to AI symbolism, and implicit semantics from contemporary deep neural network based language models. For the RFT module, we assume that any input sentence is a description of some real-world scenario, and the goal of RFT is to provide natural language explanations for those scenarios. The sentences themselves are pronoun-ambiguous (section 2.2), setting the stage for the downstream pronoun disambiguation task.

We implement RFT using generative language models that are specialized for writing summaries, so that we take advantage of their summarization capacity to generate desired explanations. Initially, with a small amount of human-provided natural language explanations, the RFT language model learns to generate likewise explanations on its own. After that, the language model uses its self-generated explanations to continue to teach itself to generate better and better explanations.

In principle, whenever the human user observes a decrease in the qualities of the model's self-generated explanations, the human has the choice to provide additional guidance to help the model get back on the right track. In our experiments, however, we have only provided the initial human explanations.

RFT can also be thought of as an instantiation of the *Learning-to-Learn* idea that permeates Meta-Learning Finn et al. (2017); Javed & White (2019); Beaulieu et al. (2020). Every time the RFT language model accepts its previous outputs as new inputs for another round of fine-tuning, the underlying gradient descent algorithm enables the model to reflect on the mapping that it had previously learned and prepare to improve the mapping. This process can repeat for any specified number of loops. Consequently, we call it *recurrent*.

2.1 Implementation Details

For experimental purposes, we administered three stages to allow for probing into RFT. To initialize, we selected 224 sentences from the Definite Pronoun Resolution (DPR) dataset Rahman & Ng (2012) and manually wrote a corresponding English explanation for each sentence. The three stages are as follows:

- To reach the first stage, we fine-tuned HuggingFace's off-the-shelf T5-Large Raffel et al. (2019) based text summarization model training pipeline[2] on the initial 224 data points, which are pairs of sentences as input texts and corresponding human-provided explanations as input labels. The specific T5 summarization model we used was T5-for-Conditional-Generation.
- To reach the second stage, we first used the previously fine-tuned T5 model to generate explanations for all the remaining sentences in the DPR dataset. Now that we have an explanation for each of the $1,886$ sentences in the DPR dataset, we fine-tuned the model for the second time on all $1,886$ data points. To clarify, for the 224 sentences in the initial step, we used the human-provided explanations; for the other sentences, we used the model-generated explanations.

[2]Code base is publicly available at `https://github.com/huggingface/transformers/tree/main/examples/pytorch/summarization`.

- To reach the third and final stage, we first used the previously fine-tuned T5 model to generate explanations for all $9,248$ WinoGrande-debiased training sentences and $1,267$ WinoGrande development sentences Sakaguchi et al. (2021) — a total of $10,515$ sentences[3]. To do so, the blank of each sentence was filled with the correct candidate. Now that we have explanations for all $10,515$ sentences in the WinoGrande-debiased dataset, we fine-tuned the model for the third time on all $10,515$ data points. This time, all explanations are model-generated.

2.2 Probing Examples

Provided that the aim of RFT is to generate natural language explanations that resemble commonsense knowledge, it makes sense that, instead of numerically evaluating the outputs of RFT via metrics, we probe the natural language explanations and make human judgments on how good they are.

Table 1 shows some probing examples that compare the qualities of human (if applicable), plain T5 model, and recurrently fine-tuned T5 model explanations.

The infection spread throughout the building because [it] was airborne.	
Human	Airborne viruses can spread quickly in ventilated in-door areas.
Plain	The infection spread throughout the building because it was airborne.
RFT	Airborne viruses can spread quickly in a building.
The infection spread throughout the building because [it] was ventilated.	
Human	Airborne viruses can spread quickly in ventilated in-door areas.
Plain	Infection spread because the building was not ventilated.
RFT	Ventilation can spread infection throughout a building.
The smoke spread through the building because [it] was caught on fire.	
Human	—
Plain	Smoke spread through the building because it was on fire.
RFT	Smoke is created when a building is on fire.
The smoke spread through the building because [it] came from the fire.	
Human	—
Plain	Smoke spread through the building because it came from the fire.
RFT	Smoke from a fire can spread quickly in a building.

Table 1: Probing examples RFT. The top twin sentences were taken from the DPR dataset, so we could provide initial human-provided explanations. The bottom twin sentences were provided by us at probing time; note that they do not have human-provided explanations, because they do not appear in any dataset. **Plain** means the default T5-for-Conditional-Generation text summarizer taken directly from the HuggingFace library without fine-tuning of any kind. **RFT** stands for our Recurrent Fine-tuning approach. Also note that the "[]" in the sentences are added to the table to clarify the pronouns of interest; they do not appear in the datasets.

[3]As we will discuss in section 3.1, we use the WinoGrande-debiased training set as our training-validation set with $80\% - 20\%$ split, and the WinoGrande-debiased validation set as our test set.

When probing, we looked for two qualities:

- Without having to think about any specific context, RFT's generated explanations should look commonsensical. If an explanation is nonsense, then any context seems irrelevant.
- Because an input sentence provides specific contexts, we ask that its corresponding explanation should appear relevant to the sentence itself. We think this is warranted because much of our human interpretation of things relies on the appearance of said things. Any attempt at formalizing the notion of *interpretability* must, in turn, answer the question of how interpretable such formalizations are.

In addition to these qualities, we also honored the following suggestions for writing initial explanations of aforementioned pronoun-ambiguous sentences:

- An explanation should be one-sentence long and should make it clear how people would resolve the ambiguous pronoun. For example, to resolve the "it" in "The bee landed on the flower because it had/wanted pollen," a human user may write "Bees like pollen/Flowers have pollen" as explanation.
- An explanation should hint at either accepting right answers or rejecting wrong answers, or both. For example, the aforementioned explanations hint at accepting right answers for the aforementioned twin sentences. Another plausible explanation is "Bees like pollen *and* flowers have pollen," combining both hints.

For all three fine-tuning stages, we tried fine-tuning the T5 summarization model with both batch size of 4 and batch size of 8. All fine-tuning steps took place for 5 epochs, and we did not observe significant differences in the qualities of generated explanations. However, we observed that the model sizes increased proportionally to the data size utilized for the stages. Table 2 shows the sizes of all RFT models after all stages:

1st stage (224)	batch$=4$	2.8G
	batch$=8$	2.8G
2nd stage $(1,886)$	batch$=4$	20G
	batch$=8$	11G
3rd stage $(10,515)$	batch$=4$	127G
	batch$=8$	127G

Table 2: RFT model sizes. Note that for the 3rd stage, $10,515 = 9,248$ (WinoGrande-train-debiased) $+1,267$ (WinoGrande-dev).

These results suggest that the T5 summarization model can learn to analogize and produce explanations consistent with what we expect. In both twin sentence pairs, we see that the plain model simply regurgitates the original sentences, lacking efforts in resolving the ambiguous pronouns and confusing the sentence semantics with their negative counterparts. The RFT model, on the other

hand, learns to associate the correct entities with their local contexts in the initial sentences and carries this ability over to new sentences.

In general, we observed that the qualities of the generated explanations did not always improve as more iterations of RFT was done. Namely, explanations generated by the second-stage models frequently looked better than those by the third models. We suspect that introductions of sufficiently different inputs during later iterations can make the RFT model forget what it had learned during earlier iterations. This model forgetting issue and the model size issue shown in Table 2, together, are reminiscent of similar issues discussed — and tentative solutions proposed — in such Meta-Learning literature as Finn et al. (2017), Javed & White (2019), and Beaulieu et al. (2020).

3. Task Specialization

As shown in Figure 1, the task specialization module consists of only an encoder and a classifier. The encoder learns abstractions of concatenations of sentences and explanations, where the sentences are the same ones as input to RFT, and the explanations are outputs of RFT. The classifier then uses the abstractions to determine the correct pronoun disambiguation decisions. By keeping the task specialization module simple, we can better investigate how much direct impact the RFT-generated explanations have on the performance of this module on downstream tasks.

To train the classifier, we adopt a very similar data preprocessing method to the one in Sakaguchi et al. (2019), where we replace the ambiguous pronoun in each sentence by its correct candidate, and then delimit all the tokens after the candidate and treat them holistically as *semantic context* that is consistent with the correct candidate.

3.1 Implementation Details

We had chosen to evaluate ProGeneXP on the WinoGrande Sakaguchi et al. (2021) dataset[4], as the downstream pronoun disambiguation task. Just like in Sakaguchi et al. (2021), we also set up our experiments as binary classification, where we train ProGeneXP to find the correct pronoun coreference resolution out of two candidates.

For the encoder, we took off-the-shelf pre-trained BERT Devlin et al. (2019) and RoBERTa Liu et al. (2019) language models, publicly available from HuggingFace[5], and fine-tuned[6] them for classification purposes. For the BERT models, we utilized both the "cased" and the "uncased" versions, meaning that the model was pre-trained on cased and uncased English texts, respectively. For input data, we utilized two different versions of the WinoGrande-debiased training set of $9,248$ sentences: plain sentences, and sentences appended with corresponding RFT-generated explanations. In total, we experimented with the following six versions of encoders:

- BERT cased, fine-tuned with explanations
- BERT cased, fine-tuned without explanations

[4]All WinoGrande data are publicly available at `https://github.com/allenai/winogrande`.
[5]All pre-trained Transformer-based models, such as BERT and RoBERTa, are publicly available at `https://github.com/huggingface/transformers`.
[6]Note that this fine-tuning is not to be confused with the fine-tuning discussed in section 2.

- BERT uncased, fine-tuned with explanations
- BERT uncased, fine-tuned without explanations
- RoBERTa, fine-tuned with explanations
- RoBERTa, fine-tuned without explanations

For the classifier, we only utilized a linear layer with gradient update, so as to make direct use of the encoder's abstractions without further manipulating them in any way. Because the embedding dimension for both BERT and RoBERTa is 768, we set the classifier's input dimension to also be 768. The output dimension is 2, because of binary classification.

For the data, we used the $9,248$ WinoGrande-debiased training sentences as our training-validation set, and the $1,267$ WinoGrande-debiased validation set as our test set. We randomly split the training-validation set into 80% training and 20% validation data, for each epoch.

In total, we fine-tuned our encoder-classifier stack for 20 epochs with a batch size of 32 and tested the stack both with and without RFT-generated explanations.

3.2 Preliminary Results

Table 3 shows our preliminary results. For baselines, we included the coin-flip probability of 50% as well as the WinoGrande baseline shown in Sakaguchi et al. (2021).

Train / Test	BERT (cased)		BERT (uncased)		RoBERTa	
	–	with descs	–	with descs	–	with descs
Baseline (coin flip)	0.5					
Baseline (WG)	–	–	0.658	0.649	0.793	0.791
WG-valid	0.867	0.859	0.863	0.852	0.855	0.856
WG-valid with descs	0.840	**0.867**	0.828	**0.858**	0.844	**0.862**

Table 3: Test results for all 6 fine-tuning settings tested on 2 different test settings. "WG" stands for WinoGrande. The WinoGrande baselines are taken directly from Table 3 of Sakaguchi et al. (2021). Admittedly, however, the WinoGrande baselines were obtained on the test set, using the Area Under the Curve (AUC) method.

Albeit simple, our ProGeneXP approach yields well above average performance across the board, suggesting that the RTF-explanations, while rendering a form of transparency into the underlying language models, can also help improve performance on downstream tasks. This makes us question the long-held idea that there is a necessary trade-off between transparency and performance.

We also observe that our results are very close to each other; we suspect this may be because ProGeneXP had successfully exploited underlying statistical biases in the WinoGrande-debiased data. For future work, we believe that more training time and more ablations across the board will help with not only improving overall performance, but also further distinguishing the individual performances of different language models.

4. Related and Future Work

In terms of architecture, ProGeneXP both resembles and contrasts the Self-Talk unsupervised framework (Shwartz et al., 2020). The clarifications in Self-Talk are similar in spirit to ProGeneXP's RFT-generated explanations. However, in Self-Talk the clarifications are generated by prompting language models with questions or prefixes, whereas ProGeneXP seeks initial and interpolating human-provided explanations. We believe that ProGeneXP offers the necessary human guidance, modularity, and operational flexibility for asking language models to elicit their implicit semantics as explicit explanations. Also, Self-Talk achieved roughly coin-flip performance on WinoGrande validation (dev) set, whereas ProGeneXP performs much better, as shown in Table 3.

Commonsense assertions from knowledge bases such as ConceptNet (Speer et al., 2017) and Atomic (Sap et al., 2019) were incorporated in several Self-Talk experiments, whereas ProGeneXP currently does not include any knowledge base. With relative ease, however, commonsense assertions and embeddings can be added to the RFT module and the task encoder. We expect to do so in our future work, bringing together language model transparency and commonsense knowledge.

Regarding commonsense, Kocijan et al. (2022) suspect that contemporary language models are so powerful at exploiting statistical patterns in (even debiased) datasets that they can achieve human-level performance without doing human-like commonsense reasoning. Today as AI communities refresh their interests in transparency and interpretability, it is worth thinking of Winograd Schema related benchmarks as aspiring for both performance and transparency. We also note such works as COMET (Bosselut et al., 2019; Hwang et al., 2020) and RetroGAN (Colon-Hernandez et al., 2021), which leverage language models to learn and generate commonsense assertions.

Throughout this paper, we extensively used the term, "explanation," specifically natural language explanations. Admittedly, we do not provide a formal definition of "explanation," as we believe that machine-generated explanations should look like commonsense to even the laypeople, and it is challenging to formalize commonsense. Whichever forms explanations take for the time being, we honor the performance of state-of-the-art black-box models and hope to implement explanation-based user interface with such models. For the future, however, we think of explainable or explanation-based, black-box models as stepping stones to interpretable, white-box models whose internal processes are understandable to people from the ground up, as stated in Rudin (2019). We believe it is urgent to begin building such interpretable models.

5. Contributions

- We explained details of our low-cost general-purpose computational framework, which leverages human-provided explanations to instruct large language models to generate transparent commonsense-like explanations, and uses such explanations to improve performance on a downstream pronoun disambiguation task.

- We showed that transparency and performance can be improved together, thus undermining the credibility of the long-held idea that there is a necessary trade-off between them, and reassuring AI researchers that pursuing transparency may pay off sooner than they had expected.

Acknowledgment

We acknowledge the gracious support of our sponsors. Xin's work is supported by the Learning, Information, and Signal Processing (LISP) group at Boston University. Lieberman's work is sponsored by the U.S. Air Force Office of Scientific Research (AFOSR) and U.S. Defense Advanced Research Projects Agency (DARPA). Colon-Hernandez's work is sponsored by the Media Lab Consortium Funding at Massachusetts Institute of Technology.

References

Beaulieu, S., Frati, L., Miconi, T., Lehman, J., Stanley, K. O., Clune, J., & Cheney, N. (2020). Learning to continually learn. *ECAI 2020 - 24th European Conference on Artificial Intelligence, 29 August-8 September 2020, Santiago de Compostela, Spain, August 29 - September 8, 2020 - Including 10th Conference on Prestigious Applications of Artificial Intelligence (PAIS 2020)* (pp. 992–1001). IOS Press. From https://doi.org/10.3233/FAIA200193.

Bosselut, A., Rashkin, H., Sap, M., Malaviya, C., Celikyilmaz, A., & Choi, Y. (2019). COMET: Commonsense transformers for automatic knowledge graph construction. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (pp. 4762–4779). Florence, Italy: Association for Computational Linguistics. From https://aclanthology.org/P19-1470.

Colon-Hernandez, P., Xin, Y., Lieberman, H., Havasi, C., Breazeal, C., & Chin, P. (2021). RetroGAN: A cyclic post-specialization system for improving out-of-knowledge and rare word representations. *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021* (pp. 2086–2095). Online: Association for Computational Linguistics. From https://aclanthology.org/2021.findings-acl.183.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *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)* (pp. 4171–4186). Minneapolis, Minnesota: Association for Computational Linguistics. From https://aclanthology.org/N19-1423.

Finn, C., Abbeel, P., & Levine, S. (2017). Model-agnostic meta-learning for fast adaptation of deep networks. *Proceedings of the 34th International Conference on Machine Learning* (pp. 1126–1135). PMLR. From https://proceedings.mlr.press/v70/finn17a.html.

Hwang, J. D., Bhagavatula, C., Bras, R. L., Da, J., Sakaguchi, K., Bosselut, A., & Choi, Y. (2020). (comet-) atomic 2020: On symbolic and neural commonsense knowledge graphs. *AAAI* (pp. 6384–6392).

Javed, K., & White, M. (2019). Meta-learning representations for continual learning. *Advances in Neural Information Processing Systems*. Curran Associates, Inc. From https://proceedings.neurips.cc/paper/2019/file/f4dd765c12f2ef67f98f3558c282a9cd-Paper.pdf.

Kocijan, V., Davis, E., Lukasiewicz, T., Marcus, G., & Morgenstern, L. (2022). The defeat of the winograd schema challenge. From https://arxiv.org/abs/2201.02387.

Liu, Y., et al. (2019). Roberta: A robustly optimized bert pretraining approach. From https://arxiv.org/abs/1907.11692.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P. J. (2019). Exploring the limits of transfer learning with a unified text-to-text transformer. From https://arxiv.org/abs/1910.10683.

Rahman, A., & Ng, V. (2012). Resolving complex cases of definite pronouns: The Winograd schema challenge. *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning* (pp. 777–789). Jeju Island, Korea: Association for Computational Linguistics. From https://aclanthology.org/D12-1071.

Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, *1*, 206–215. From https://doi.org/10.1038/s42256-019-0048-x.

Sakaguchi, K., Bras, R. L., Bhagavatula, C., & Choi, Y. (2019). Winogrande: An adversarial winograd schema challenge at scale. From https://arxiv.org/abs/1907.10641.

Sakaguchi, K., Bras, R. L., Bhagavatula, C., & Choi, Y. (2021). Winogrande: An adversarial winograd schema challenge at scale. *Commun. ACM*, *64*, 99–106. From https://doi.org/10.1145/3474381.

Sap, M., Le Bras, R., Allaway, E., Bhagavatula, C., Lourie, N., Rashkin, H., Roof, B., Smith, N. A., & Choi, Y. (2019). Atomic: An atlas of machine commonsense for if-then reasoning. *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*. AAAI Press. From https://doi.org/10.1609/aaai.v33i01.33013027.

Shwartz, V., West, P., Le Bras, R., Bhagavatula, C., & Choi, Y. (2020). Unsupervised commonsense question answering with self-talk. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 4615–4629). Online: Association for Computational Linguistics. From https://aclanthology.org/2020.emnlp-main.373.

Speer, R., Chin, J., & Havasi, C. (2017). Conceptnet 5.5: An open multilingual graph of general knowledge. *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence* (p. 4444–4451). AAAI Press.