CSC-791 Term Paper

Luke Roosje
Department of Computer Science, NCSU
leroosje@ncsu.edu

Vinayak Dubey
Department of Computer Science, NCSU
vdubey2@ncsu.edu

***Abstract*—Pronoun Disambiguation is an important challenge in Natural Language Processing. Specifically, the Winograd Schema Challenge is a difficult pronoun disambiguation problem even touted as an alternative to the Turing Test. Over the years since its conception, the challenge has had results with topping accuracy in the early 60s. We propose a novel method to solving this challenge by approaching it as a question answering problem. Our approach takes inspiration from Visual Question Answering common in the vision area and we propose to adapt the same to solve this tough task. Our results validate our hypothesis.**

***Index Terms*—NLP, pronoun disambiguation, coreference resolution, Winograd Schema Challenge**

I. Introduction

The Winograd Schema Challenge [8] was proposed as an alternative to the Turing Test. It poses a set of multiple-choice questions based on two consequent sentence clauses. For example: noitemsep

- The town councilors refused to give the demonstrators a permit, because they feared violence.
- Snippet: **they** feared violence
  - Answer-1: The Town Councilors
  - Answer-2: The Demonstrators

Any human, with a fair bit of knowledge about the English language, will be able to answer the above question correctly. However, for a machine to do that, it must understand the structure of the sentence as well as the semantic relationships between the words (for example: the understanding that the verb feared is associated with the noun councilors and not demonstrators).

We will first introduce some earlier efforts made to solve this particular challenge as well as other pronoun disambiguation challenges, and then introduce two hypotheses inspired by those efforts.

II. Related Work

Trinh et. al. [14], presented a solution which involved the use of pre-trained language models to predict the probability of the second clause of a Winograd sentence with the pronoun replaced with one of the options, given the first clause of the sentence.

Ionita et. al. [6], in their attempt to solve Kaggle's GAP Challenge, implemented BERT [4]. In their work, they mention the use of hand-crafted features to help with prediction. These features include the predictions of a Multi Layer Perceptron trained on ELMo [11] embeddings, predictions from BERT [4] among others.

A premier challenge in the Computer Vision field is the Visual Question Answering Challenge [2], which poses questions based on an image. A vanilla solution for this [7] involves feature extractions of the knowledge base (the image in this case), the question and then combining/concatenating/multiplying the two to form inputs for a classifier.

III. Hypothesis I

NLP techniques can solve the Winograd Schema Challenge when the problem is posed as a Question-Answering problem

This approach takes inspiration from the Visual Question Answering [2] challenge and its baseline solutions [7].

A. Our Approach

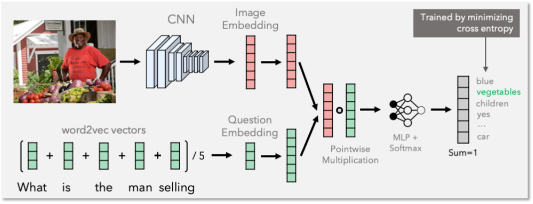

Fig. 1. A Vanilla-VQA Model; source: [7]

1) Visual QA, a brief overview: This is a challenge where a question based on a knowledge base (image) is posed to the system, and it must answer by using the knowledge base. As shown in Figure 1, a vanilla model for VQA extracts image and question representations, projects them to the same number of dimensions (using fully connected layers) and then combines them to form inputs for a classifier which predicts an answer. We take inspiration from this model to design our first hypothesis.

2) Our Model and it's Pipeline: We propose to reshape the pronoun disambiguation problem into a question answering problem instead. For example: in the example presented in the Introduction section, we will reference the snippet in the form of a question: "Who feared violence?".

Our model will then consist of the following layers:
Input Layer noitemsep

- **Dataset:** Rahman et. al. [12] in their attempt at the Winograd Schema Challenge, prepared a data set consisting of

943 Winograd-like sentences. We will leverage it as our training set.
- **Question Generation:** We will use the NLTK parse package to convert strings like ”it is too big” to ”what is too big”. We understand that these productions will not always be correct and we hereby disclaim that we will be putting in some manual effort to polish the questions.
- **Extra Answer Module:** Instead of predicting the possibilities of the two answers, we will also explore a different approach by adding another module of the answer itself as the input. In this case, we will be predicting the probability of the answer itself and not 2 class’s probabilities.
- **Tokenization and Stop Words:** The NLTK tokenization package will be appropriately leveraged, and we will use a custom list of stop words to be removed from all the three modules: the knowledge base, the question, the answer. We will be avoiding the NLTK stopword collection because it consists of too many words that form the most important parts of a Winograd sentence.

Representation Layer noitemsep

- Since sentence structure is important in this area, we will be employing TensorFlow’s Universal Sentence Encoder technique [13] for embedding all the three input modules: the knowledge base, the question and the answer.
- We will also explore BERT embeddings. It has been shown that it is possible to extract embeddings from BERT. It is to be noted that these embeddings will be sentence embeddings and not word embeddings. It will be performed through [5].

Merge Layer noitemsep

- In Visual QA pointwise multiplication is the most used method for combining the image and question representations. We will be experimenting with pointwise multiplication, pointwise addition and concatenation. This merged result, will then serve as input to the classification layer.

Classification Layer noitemsep

- We will employ the MLP and SVM APIs of the sklearn library to predict the probability of the the answer being correct (or not). We will employ a softmax function to receive the output.

3) Metrics: Since we are experimenting with 2 different kinds of questions (See Section: 6.1.1), we will be providing results for both of them as well as a combined dataset. For each of these, following are the metrics:

- All previous submissions of the Winograd Schema Challenge have used accuracy as the sole measure of performance. We plan to do the same.
- Along with that, we will also be reporting a confusion matrix for the same.

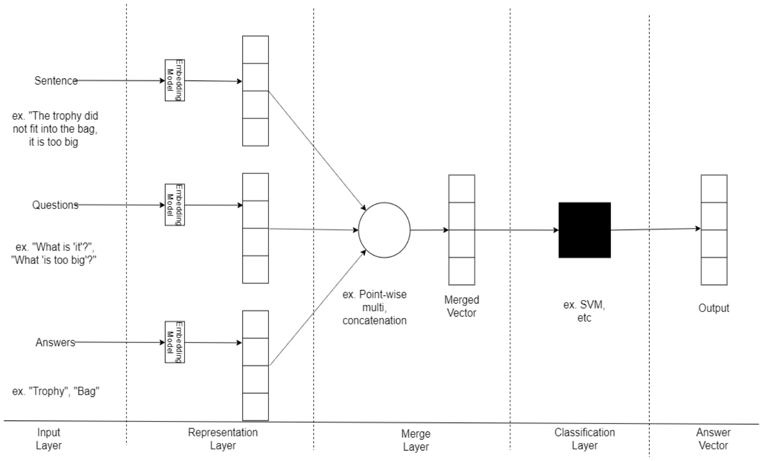

Fig. 2. Our solution’s architecture

IV. Hypothesis II

A. The Hypothesis Statement/Research Question

Can hand crafted features help in solving the Winograd Schema Challenge?
This approach takes inspiration from the use of hand crafted features in [6].

B. Our Approach

1) Our Model and it’s Pipeline: We plan to exploit hand crafted features for this hypothesis. [6] used features such as results from a MLP trained on ELMo embeddings among other features. We plan to do the same, adding a few more features. We will use the same dataset as cited in the first hypothesis [12].

Input Layer

- **Previous Prediction:** Since [14] currently holds the top spot on the leader-board, we will use their prediction as one feature.
- **Tokenization and Stop Words:** The NLTK tokenization package will be appropriately leveraged, and we will use a custom list of stop words to be removed from the all the three modules: the knowledge base, the question, the answer.
- **POS and Similarity:** The Winograd dataset’s XML format divides the example into two parts depending on the position of the target pronoun. We will use this to split the the input into two parts. And then, each part will be processed to identify POS’s (through the NLTK pos package) such as Verbs, Nouns and Adjectives. Similarity(using the NLTK package, wordnet) will be calculated between each possible noun and the verbs and adjectives in the second sentence. This forms 4 features: Noun1-(Verb1 + any other verb), Noun1-(Adjective1 + any other adjective) and similarly for Noun2. While this might not sound very intuitive, but looking at the dataset, one can find many examples such as: ”Woman smiled

at the girl, while stretching her back". Looking at such an example, we are more inclined towards the fact that a woman probably has a greater need to stretch her back than a girl(also reflected by the similarity measure returned by NLTK's wordnet $-->$ .33 vs .30).

- **Gender:** We add two more features that inform whether the genders of a pronoun and the noun match. NLTK provides a tutorial [9] on how to train a classifier to do the same. Value will be 1 for same gender, 0 for different genders and -1 for genderless entities.

Classification Layer

- The 7 features will serve as input to a MLP or SVM (depending on which performs better on the validation set) to predict the correct noun for the pronoun.

2) Metrics: All previous submissions of the Winograd Schema Challenge have used accuracy as the sole measure of performance. We plan to do the same.

V. Implementation

A. Hypothesis I: Adapting VQA

Since the proposal, our main hypothesis has seen considerable progress. We have been able to implement it and get results.

1) Implementation Details:

Input Layer The data at the input layer of our project has been generated using Python scripts on the data-set we mentioned above. Both the training and testing data are well formatted, which has made this task very straightforward. The data has been placed into seperate six-column .csv files, to make generating embeddings streamlined.

Details

- **Knowledge Base:** The knowledge base is a cleaned version of the original sentence in the dataset. It is made all lowercase, and has the stopword "the" removed. We are still considering removing other stopwords, but "the" was, by far, the most prominent. Other possible stopwords like "and" or "which" may be removed in the future, but they occur infrequently, so more testing is required before we decide.
- **Questions:** There are two kinds of questions generated for each piece of data in our sets. If the original knowledge base is: *"The bee went to the flower, it had pollen."*, then our two forms of questions are:
  - **Form-1:** *What does $<$ pronoun(here, it) $>$ refer to?*
  - **Form-2:** *What had pollen?*

  Each of these questions has its own entry in the CSV, with the rest of the data points (KB, Options, answer) being the same. We included two questions for each data point so that our training data set is much larger, and our classifier will be stronger. While we could have implemented the two questions into one csv entry, processing this data later would not be as straightforward and easy to read. We believe this ease of processing and scalability is more important than the somewhat negligible extra storage requirement.

  Further, asking both questions of the test data gives us an opportunity to analyze our results on a deeper level:
  - Will one question provide better results than the other?
  - Will they perform equally?

  These questions will be answered with further testing.
- **Options:** Options are a formatted version of the answer choices available to the classifier. By including their embeddings, we can add extra strength to the classifier using the "closed-choice" scheme described earlier. The options are formatted so that they can be processed by a new embedding system that we found, which will be described later.
- **Actual Answer:** The actual answer has two entries in each csv row. One of them is noun itself, which is there to aid the reader's intuition. The second entry is the label: a 0 or a 1, depending on which option is the answer to the question.

2) Representations: Very recently, [15] and [3] proposed a model for sentence embedding specifically suited for question answering. This technique involves:

- A question embedding, which returns an embedding for the question.
- A response embedding, which returns an embedding for the answer given a context (in our case, the knowledge base)

We deviated from our initial intention of using BERT embeddings [5] and instead went for the above stated embedding techniques.

It is to be noted that the above stated embeddings do not have an API exposed to access the embeddings of the response contexts. However, the knowledge base is a major input in our pipeline. Therefore, we went for Tensorflow's Sentence Encoding Technique in order to generate embeddings for the knowledge base as well.

3) Merge: Until now, for our embedding merge layer, we have experimented with concatenation of the three generated embeddings (question, answer, knowledge-base). As we progress, we will experiment with more techniques.

4) Classifier: We experimented with two classifiers:

- A SVM Linear Classifier [10]
- A Multi-Layered Perceptron [10]

The Multi-Layered Perceptron was able to provide an increase in our accuracy by at least 3 points, therefore, we plan to include that in our final implementation.

B. Results: Hypothesis-1

1) Form-1: **Accuracy:** 59.50%

Confusion Matrix:

Labels	Label-0	Label-1
Label-0	75	47
Label-1	68	94

2) Form-2: **Accuracy:** 53.17%

Confusion Matrix:

Labels	Label-0	Label-1
Label-0	67	57
Label-1	76	84

3) Combined: **Accuracy:** 57.21%

Confusion Matrix:

Labels	Label-0	Label-1
Label-0	138	95
Label-1	148	187

Our implementation of Hypothesis I remains mostly the same, save one change. In our original implementation, there was a bug in generating our test data. The Winograd Schema Challenge's dataset has an entry for the ambiguous phrase surrounding the pronoun, rather than the pronoun itself. This led to a misgeneration of the pronoun being resolved. By using NLTK's POS tagging, we were able to find the actual pronoun in the phrase and properly feed it into our classifier for testing. This lead to an increase in the accuracy for Form-1 questions to a point where it beats the current state of the art result. The results have been captured below:

4) Form-1 Improved: **Accuracy:** 62.33%

Confusion Matrix:

Labels	Label-0	Label-1
Label-0	84	48
Label-1	59	93

C. Hypothesis II: Using Features

To re-summarize Hypothesis-2, the idea was to use hand crafted features instead of word embeddings to solve the Winograd Schema Challenge's pronoun disambiguation problem. We had previously proposed to use features including: gender based features, similarity with other parts of speech and previous predictions.

Features leveraged:

- **Previous Prediction:** We previously proposed to use the predictions of [14] as a feature in R1 and R2. However, their results were based on an older version of the Winograd Schema Challenge. Also, we did not have the required resources to reproduce their results on the newer version (It required roughly 50 gigabytes of memory to run all their models since we also needed their predictions on our training data).
  Also, since we achieved an even better accuracy in Hypothesis-1 during this phase, we decided to use those predictions as a feature.

- **POS and Similarity:** This feature is based on certain observations made from reading the data. For example, in the case of *The bee went to the flower, it wanted pollen*, there is a clear line dividing that out of the two possible antecedent nouns (bee and flower), only one of them could *perform* the verb in the second sentence (want).
  Similarly, in another example: *The woman smiled at the girl, she was stretching her back*, we are more inclined toward the fact that a woman probably has a greater need to stretch her back than a girl.

  We have tried to capture this relationship between entities using POS tagging and similarity. Specifically, we took the two possible antecedent nouns from the first phrase and measured its similarity with the verbs and adjectives from the second phrase. This gave us 4 features: Noun1-(Verb1 +any other verb), Noun1-(Adjective1 + any other adjective) and similarly for Noun2.

- **Gender:** A huge portion of pronoun/subject pairs in day-to-day English are gendered. If only one male name has been introduced in a discourse, a pronoun ”he” is extremely likely to be referring to that male. As such, we sought to leverage these relationships to better solve our pronoun resolution problem.

  We implemented this using a classifier built through NLTK, and its premade datasets of names. The classifier was trained on these sets of names and their respective genders, and used to predict the gender of the pronoun in our training and test data, versus the answer choices. If a pronoun's predicted gender matched that of an answer, a 1 was recorded in a corresponding CSV column, and if it did not, a 0 was recorded instead. Ideally, this feature can determine that one answer choice matches gender and the other does not, which is a massive indicator of which answer is correct.

Many of the answer choices in our dataset are not proper nouns, and thus don't have a very distinct gender. However, nouns in general have gendered connotations. For example, ”doctor” may be traditionally associated with men, while ”nurse” may be traditionally associated with women. In conjunction with a gendered pronoun, this may produce favorable results.

D. Results: Hypothesis-2

Out of all the features that we provided to all the classifier models that we experimented with, all the models only gave emphasis to the ”previous prediction” feature. That is, all models provided the same accuracy and the same classification results as Hypothesis-1. None of the other features of gender, POS Similarity, etc., were given importance.
Therefore, our accuracy and confusion matrix for this hypothesis turn out to be exactly same as Hypothesis-1:

Accuracy: 62.33%

Confusion Matrix:

Labels	Label-0	Label-1
Label-0	84	48
Label-1	59	93

The above results are exactly the same as in Hypothesis-1.

VI. Results against the largest community

Our best results had an accuracy of about 62.33% on the Winograd Schema Challenge's set of pronoun resolution questions. This is an additional 12.33% stronger than a truly random guesser which seems insignificant, but actually is a strong result. When this challenge was first created in 2016, the best results achieved up to 58% accuracy.

As the Winograd Schema Challenge's leaderboards show, this result is .83% more accurate than the state-of-the-art. This is a huge step forward, and headway towards building a machine that can reliably resolve pronouns.

It is to be noted that there have been other results too of late based on attention based transformer models such as BERT($\tilde{6}5\%$) and GPT2($\tilde{7}0.7\%$). While these models are not official entries, but being the best language models at present, hold their stand. We further hypothesize that in future work, if we use such strong sequential neural models instead of the basic MLP Classifier, we will be able to achieve even better results.

VII. Limitations

Our second hypothesis is built as an extension of the first. Thus, all limitations with our first hypothesis extend into the second. Subsections will be divided as such.

A. General + Hypothesis I

- **Limitations in adapting:** One of the largest issues with adapting VQA is in the difference in knowledge bases. VQA works well because images are incredibly rich sources of information. There are huge numbers of pixels, colors, and other attributes that are leveraged to create very strong classifiers. Replacing images with sentences loses this breadth. This is countered by using sentence embeddings, which provide a huge amount of context and data about any sentence, but these are still a far cry from what can be derived from images.
- **Issues with VQA:** VQA is a relatively new technique in machine learning. There is still progress to be made in that field, so adapting it inherits the issues that remain. As these issues are resolved and the techniques are strengthened, our work will benefit and yield better results by adapting the same fixes.
- **Preprocessing:** While we preprocessed our test and training data, there may be more to improve. Normalizing the case of characters or removing more stopwords may improve our results.
- **Embeddings:** We used common and established sentence embedders to create embeddings that we fed to our classifiers. These are excellent multi-purpose tools, but there may be better, more nuanced ways, to prepare our data and make note of strong indicators within the sentence for the classifier to recognize.

B. Hypothesis II

Leveraging features is a common approach to solving problems in NLP, and this problem is no different. Features are an excellent way to highlight important aspects of a statement to a computer, but the biggest issue is their volatility. Intuitively, having more information would help something answer a question. However, when being translated into embeddings and then processed by a classifier, these patterns can be warped and altered unpredictably. Thus, each of these features can only be indicative of themselves, and not others, and should be judged accordingly. In future work, other features should be tested as well. Each of our chosen features have their own limitations as well. They will be discussed next.

Something that may benefit this hypothesis in the future is incorporating weights into all the data we feed into classifiers. Right now, data is all weighted proportionally to its number of datapoints. Thus, single data points which represent important aspects (like features) are drowned out by more numerous parts. This is evident in our work; our results after features are incorporated do not change, suggesting that they are lost.

This can be solved by constructing our final embeddings differently to give these important datapoints more presence, or by adding weights. Adding these adds a new front to our research, though. How will the correct weights be determined? What should be weighted highly? Can it be determined algorithmically or does it require manual labelling? With further work, exploration, and time, perhaps our model could be stronger and improved. Next, we address points of our project that could be explored and improved as a next step.

- **Previous Prediction:** In theory, this feature wouldn't affect the predictions on our test data. If our classifier was already going to output this answer, being told it was going to output that answer beforehand should just increase its confidence in its answers. However, when the data fed to the machine is altered with other features, the machine will not necessarily output the same answer as before. Thus, this feature takes on a new role; a stabilizer. It provides an anchor for the new machine to work off.
- **Gender:** In practice, gender isn't the huge indicator it could be on our problem set. Most questions are using un-gendered pronouns, so the gender of an answer choice doesn't matter as much.

  Further, the training data available for our gender classifier is limited to proper nouns, specifically human names. This very strict set might not scale well to proper nouns, or nouns in general. Considering a larger, more diverse, training set may yield a stronger classifier. Further, leveraging more features within this training set than just the name may strengthen our classifier.

  For future work, there are plenty of other parts of a sentence that can gendered, like verbs. Other languages may find this feature much more useful as well. Spanish, for example, has gender as an aspect of its grammar. Having it be so widespread could be exploited.

VIII. Takeaways

- One the official webpage for the Winograd Schema Challenge [1], the leaderboard shows a top accuracy of 61.5%. Our results suggest a new state-of-the-art solution for this problem, providing a .83% increase in accuracy. While this is a small increase, any increase on such a difficult problem is a significant step forward, and beating previous records is an incredible accomplishment.
- It is to be noted that these results were obtained with a Multi-Layered Perceptron Classifier with the features as word embeddings. We further hypothesize that with strong sequential neural models (for example LSTM's) we might be able to further increase our accuracy.
- Nevertheless, even with a weak classifier, our results suggest that approaching pronoun disambiguation as a Question Answering problem is a viable and competitive approach.
- This is certainly a large claim to make. In future work, we will verify and reproduce them. Further, more work must be done to verify that this is, in fact, the new state-of-the-art. This could be accomplished through a peer-review process. Also, our hypothesis needs to be tested against other pronoun disambiguation problems out there such as PDP and Google's Gendered Pronoun Disambiguation Challenge.
- Further, a .83% increase, while a small number, is a huge step forward in this context. The Winograd Schema Challenge is a notoriously difficult problem in NLP, and in the three years it has been open, only the low 60's have been reached in accuracy.
- Most importantly, this work speaks to the value of testing new approaches on open problems. Even the hardest problems have solutions, even if it hasn't been discovered yet. Testing new methods and approaches allows us to learn something new, and take steps towards these grand solutions, even if the steps are small. We hope our work inspires and allows its readers to find good takeaways about how to solve this problem.
- Pronoun resolution is an incredibly difficult problem for computers to solve, but is a central part of day-to-day speech. The Winograd Schema Challenge represents some of the most difficult parts of this difficult problem, and taking steps towards solving these aspects are incredibly useful in the larger community, and building machines that can understand day-to-day language.

IX. Conclusion

Pronoun resolution is a surprisingly difficult problem to solve. While it is very intuitive for humans, the amount of nuance in natural language that allows our brains to resolve this problem is extremely difficult for a computer to comprehend. This problem is well formalized in the Winograd Schema Challenge, which offers a collection of sentences and corresponding proper nouns, with an ambiguous phrase referring to one of them. Our work has produced significant results on this challenge, reaching 62.33% accuracy and outperforming the top performer on the Winograd Schema Challenge's leaderboard. This extra .83% over the previous best classifier is a strong step in solving this problem. While this problem has yet to be solved satisfactorily, our progress suggests that it is possible.

References

[1] Official winograd challenge webpage.

[2] Aishwarya Agrawal, Jiasen Lu, Stanislaw Antol, Margaret Mitchell, C. Lawrence Zitnick, Devi Parikh, and Dhruv Batra. Vqa: Visual question answering. *Int. J. Comput. Vision*, 123(1):4–31, May 2017.

[3] Muthuraman Chidambaram, Yinfei Yang, Daniel Cer, Steve Yuan, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. Learning cross-lingual sentence representations via a multi-task dual-encoder model. *CoRR*, abs/1810.12836, 2018.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[5] hanxiao. *bert-as-service*.

[6] Matei Ionita, Yury Kashnitsky, Ken Krige, Vladimir Larin, Denis Logvinenko, and Atanas Atanasov. Resolving gendered ambiguous pronouns with BERT. *CoRR*, abs/1906.01161, 2019.

[7] Aniruddha Kembhavi. *Vanilla VQA*.

[8] Hector J. Levesque, Ernest Davis, and Leora Morgenstern. The winograd schema challenge. In *Proceedings of the Thirteenth International Conference on Principles of Knowledge Representation and Reasoning*, KR'12, pages 552–561. AAAI Press, 2012.

[9] NLTK. *NLTK Gender Classification Tutorial*.

[10] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[11] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *CoRR*, abs/1802.05365, 2018.

[12] Altaf Rahman and Vincent Ng. Resolving complex cases of definite pronouns: The winograd schema challenge. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 777–789, 2012.

[13] Google TensorFlow. *Universal Sentence Encoder*.

[14] Trieu H. Trinh and Quoc V. Le. A simple method for commonsense reasoning. *CoRR*, abs/1806.02847, 2018.

[15] Yinfei Yang, Daniel Cer, Amin Ahmad, Mandy Guo, Jax Law, Noah Constant, Gustavo Hernandez Abrego, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. Multilingual universal sentence encoder for semantic retrieval, 2019.