Attention To Detail Is What You Might Need

Srik Gorthy srik.gorthy
Noah Shen noahshen2023
Ruth Bagley ruthbagley2022
@u.northwestern.edu

Abstract

Large Language Models (LLMs) are ubiquitous, starting from BERT and GPT to current day models such as GPT-3 and PaLM, and to the even bigger models frequently being released. However, directly using these models in practice has many problems ranging from unreliability to explainability and ethics. We propose an approach to use these models for converting natural language into first-order logic. Furthermore, this approach can be repurposed to resolve the aforementioned issues. We try to see if attention-based models can generate detailed data for extracting information given a text. That is the idea behind the name of the paper, i.e., 'Applying' Attention (Attention-based models) to Detail (Extracting detailed information in the form of knowledge triples) is what you might need (could be the way to leverage the existing LLMs to generate knowledge).

1 Introduction

1.1 Problem

Current state-of-the-art neural language models have been quite successful at question answering, but they have been criticized for their opacity, or inability to see how decisions are made, and their inflexibility with new tasks or information [2]. The symbolic approach is much more transparent but lacks the natural language ability to communicate clearly. To combat this, we are using compositional question answering, where natural language is converted to logical formats. [3]

1.2 Rationale

Using a purely symbolic approach or a purely neural approach would have some major flaws, as discussed in the prior section. Consequently, our approach seeks to combine the two types of models, using neural models to convert natural language into logical triples, which could then be used by a symbolic interpreter to answer the questions. This method would allow for transparency in the reasoning process that comes from symbolic models along with the natural language component that comes with neural models.

1.2.1 Explainability

A model built on first-order logic is answerable on multiple levels of abstraction. This allows usage of large models in complex systems with explainability and aids in efforts toward Responsible AI.

1.2.2 Increasing Model Sizes for Abstractions

"In NLP, the bigger the model, the better the results". This is becoming more and more evident in recent days. Having a high parameter deep neural network model allows it to learn more abstractions and makes the chances of an improved metric stronger. However, if our model/idea is successful, the primary reason for a bigger model would be deemed void. This allows for more research into improving models while consuming less computational power and indirectly helping in making deep learning more sustainable.

1.3 Our Approach

Our goal is to build a model which converts a given text to a set of rules given a Knowledge Base. To achieve this, we first find and preprocess an appropriate dataset (Section 2), then examine prior works in the area (Section 3). Then we train and evaluate different models (Section 4) and consider ways to expand on our work (Section 5).

2 Data

2.1 Data Sourcing

We repurposed the WebNLG Dataset[1] to convert text into predicates. We employed the mtriples or the modified triples from the Web_NLG dataset along with the text for our modeling. The data is a rich source of text and related predicates. We used the V3_en for our project.

Description	Train	Dev	Test
# Examples	13,211	1667	5713
Max tokens/text	72	60	80
Min tokens/text	3	4	3
Mean tokens/text	19.66	19.69	21.82
Max Triples/text	7	7	7
Min Triples/text	1	1	1
Mean Triples/text	2.91	2.90	3.19
# Unique Rels	372	290	227
New Rels w/ Train	0	0	40

Table 1: Analysis of the tokens/text, triples/text and the repeated relations for the 3 datasets

We also explored a variety of other datasets, but WebNLG was ultimately determined to be the most appropriate for our task.

2.2 Exploratory Data Analysis

Understanding the data helped us choose the path to take and how to transform the data as necessary. We see that the calculated statistics from the train and dev data sets are similar (see Table 1), except for the maximum lengths of the strings. The maximum number of triples for one piece of text in each of the datasets is seven. Also, we observed that the test set has 40 new predicates or relations not seen in the train or dev sets (see Table 1). This information prompted us to see the accuracy of predictions in these 40 separately and compare them with the rest after the modeling (See Section 4.3.2). This allows us to see if the model is remembering the predicates or whether it is using its pre-trained architecture's abstractions to predict these relationships.

2.3 Examples

Three examples of the original text are shown in Table 2. These specific examples were chosen as they represent the best (BLEU = 1), average (BLEU = 0.7) and worst (BLEU = 0.07) predictions after the model was built.

2.4 Cleaning of Triples

We used regular expressions to convert the text in the triples to a more uniform form for the training of the model. Specifically, we converted all the text in triples to lower case, converted metrics to shorthand(_in meters to _m), etc.

3 Related Works

Compositional question answering has been explored in other work; in Liang, et al (2011), a probabilistic model was created to map questions into a logical form, which could then be resolved into an answer. The particular challenge being addressed in the paper was avoiding costly manual annotation of data for supervised models while grasping the more complex linguistic phenomena, such as superlatives, that unsupervised models struggle with. A new semantic form is developed and has good accuracy on the GEO and JOBS datasets (91.1% and 95% respectively). [3] Our goal is also to eventually have a model that converts from natural language to a logical form to answer questions, but we chose to use the form of triples because then we can import them into pre-existing knowledge bases to do the logical reasoning.

Semantic parsing for logical question answering using weak supervision has also been implemented in Liang, et al (2017). In this case, triples were also used, but the focus was on succeeding using only weak supervision and reinforcement learning. [5] This model implements knowledge bases in the way we would in future work, but use a sequence-to-sequence model with RNNs to convert between natural language and logical forms, whereas we are using transformers.

Damonte and Monti (2021), like prior works, note the difficulty of training semantic parsing systems for question answering given the small amount of properly annotated data available, and the expense of acquiring such data. Their goal was to use Multi-Task Learning to train on several smaller datasets, rather than needing to build new larger datasets. This model, like in Liang, et al [5], uses the sequence to sequence models for semantic parsing. [6] We have not yet run into a problem with a lack of data and have been using supervised learning for our training, but it appears to be a well-recognized problem in the development of semantic parsing.

The dataset we chose, WebNLG, has been used in the past primarily for Data-to-Text Generation, as was the intended purpose. One notable work using this dataset was Rebuffel, et al, where the goal is creating the QuestEval metric for evaluating summaries of text by asking and answering questions. The metric proved comparable to BERTScore and better than BLEU. [7]

Original Text	Original Triples
Nie Haisheng born on 10/13/1964 is a fighter pilot.	['Nie_Haisheng | birthDate | 1964-10-13', 'Nie_Haisheng | occupation | Fighter_pilot']
MotorSport Vision is located in Fawkham.	['MotorSport_Vision | city | Fawkham']
Espen Lind is a writer of the song Mermaid by the band Train	['Mermaid_(Train_song) | writer | Espen_Lind']

Table 2: Examples of the original text and triples from the test set.

Model	Train Loss	BLEU	Accuracy
gpt2	0.340	0.6820	0.1258
t5-small	0.0814	0.7336	0.3359
t5-base	**0.0274**	**0.7474**	**0.3938**

Table 3: Metrics for the models

4 Modeling

4.1 Architecture

The existing models which use WebNLG are used for Natural Language Generation tasks, such as generating natural language text from structured data. As our problem is the reverse, we repurposed the same to build predicates from the text. The model architecture is shown in Figure 1.

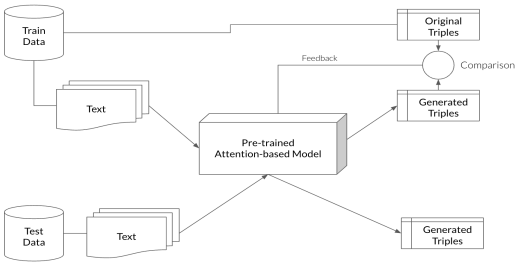

Fig 1: The proposed model architecture

4.2 Models

For our experiments, we built GPT-2 [1] and two t5 variants (t5-small and t5-base)[2]. The code can be found in this GitHub Repository [3]

The t5 models were run for 4 epochs each with a batch size of 8. Both the source and target max token length was set to 128. Training time was approximately 2 hours for t5-small and 4 hours for t5-base, running on the Google Colaboratory using GPU (K80, P100, T4), 8 vCPUs, and High-RAM(30GB+24GB).

[1]https://huggingface.co/docs/transformers/model_doc/gpt2
[2]https://huggingface.co/docs/transformers/model_doc/t5
[3]https://github.com/srikg-msai22/SSLM-group9

4.3 Results

The results of the models were evaluated using training loss, BLEU score, and accuracy. Accuracy was determined by looking at the percent of triples in the correct answer that had exact matches in the predicted answer, regardless of the ordering of the triples. In all three metrics (as seen in Table 3), t5-base performed better than t5-small, though there was the biggest difference in the training loss. From these results, it's clear that t5-base fits the data better than t5-small after training, and both fit much better than GPT-2.

Looking only at the BLEU scores, the models seem quite successful in terms of their results, with both t5 models achieving scores greater than 0.73, and all models scoring at least 0.68. Looking at Fig. 2, one can see that the most common BLEU scores by far were 1 or very close to 1. Typically perfect scores indicate over-fitting, but that is true mainly for natural language, which is what BLEU is meant to evaluate. Since we are not generating true natural language and there is one correct answer, we also evaluated the accuracy of the models.

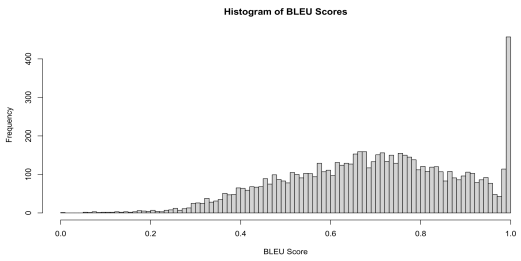

Fig 2: The histogram of the BLEU scores

4.3.1 Accuracy

Accuracy is an important metric in this instance because the produced text should be in a logical form that a knowledge base could understand. For that to work, the triples need to be entirely correct. The average accuracy for the predictions is shown in Table 3, where t5-base shows improvement over t5-small, and both markedly better than GPT-2.

Model	BLEU	Accuracy
gpt2	0.6534	0.0053
t5-small	0.5656	0.0760
t5-base	**0.5948**	**0.1254**

Table 4: Mean BLEU scores and accuracy for the subset of test dataset with relations not seen in training

An unexpectedly common problem was creating entity names, like changing the spellings; for example, one correct tuple had "Mermaid", while the predicted tuple had "Meermaid". Other times the model would hallucinate and replace words; for example, "The_Fellowship_of_the_Ring" became "Berry_of_the_Ring" in one prediction. Primarily because of these two phenomena, 43% of the predicted sets of triples for t5-base had no exact matches to the actual triples. In addition, the predictions are usually internally consistent (e.g. if one triple in a prediction has "Meermaid", the others will too), but not consistent overall; "Meanmaid", "Amermaid", and "Mean_maid" all appear in different predictions as variations of "Mermaid". This is cause for concern because each reference to a particular entity in a knowledge base needs to be written in the same way.

However, there are some significant successes. For instance, the text "MotorSport Vision is located in the city of Fawkham, UK" had an original tuple of ['MotorSport_Vision | city | Fawkham'], while our model generated ['Motors_Sport_Vision | location | Fawkham', 'Fawkham | country | United_Kingdom']. This is another example of the model's creative entity naming, but more importantly, the predicted tuples have more information than the original.

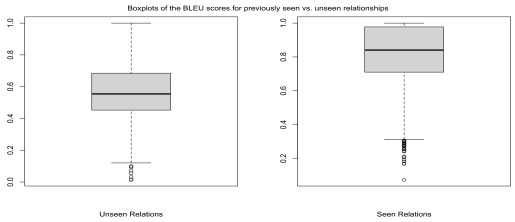

Fig 3: Boxplot of BLEU scores for Test Triples with unseen relations in train vs previously seen relations in train

4.3.2 New Relations in the Test Data

One consideration previously mentioned was how the models perform when faced with previously unseen predicates. As can be seen in Fig. 3, the examples with unseen predicates or relations are notably lower than the examples containing only relations previously seen in training. One important concern is that GPT-2 scored 0.65 on BLEU, but had an accuracy of less than 0.5%, indicating that BLEU is not a particularly reliable metric in this task. This was further verified by examining some of GPT-2's predictions individually.

Additionally, in Table 4 we can see that the mean BLEU scores and the accuracy of the subset improved from t5-small to t5-base. Based on this we can infer that the LLMs are not just memorizing the relations in the data set, but are able to leverage the existing word abstractions in them to come up with better triples. If we use a better model, better architecture, or an ensemble of multiple models, this difference may be bridged.

5 Future Work

There are several directions one could go to take this experiment further. For instance, t5-base was an improvement over t5-small, so one could try training on even larger models or ensembles of models to see even more improvement.

Additionally, some improvements need to be made to entity recognition to aid in disambiguation. For example, in example 1 of Appendix 7.2.2, we can see that the ambiguity in the word "English" as a language conflicted with the "English" as a nationality in the film title 'English Without Tears'.

6 Conclusion

To convert natural language into logical triples, we trained three models (t5-small, t5-base, and GPT-2) on an adapted version of the WebNLG dataset. T5-base was the most successful, indicating that a larger model might do even better. One significant discovery is the inadequacy of BLEU scores in evaluating language in logical forms; exact matching accuracy was more reliable.

In addition, it is important to note that these models have predicted the text similar to a translation rather than extracted knowledge/symbolic logic out of the text. However, to continue the "Attention-to-Detail" approach presented in this paper, we think the following ideas have potential:

- Building validation metrics based on the answers which can be built through the generated triples
- Creating and implementing loss functions with a human in between and augment-

ing them with feedback extrapolation to improve Human-Neural_Model inter-understandability

- Evaluating various models on the quality and types of triples they generate to use as a proxy on how much knowledge is being abstracted by them

References

[1] A. Moryossef, Y. Goldberg, I. Dagan, "Step-by-Step: Separating Planning from Realization in Neural DatatoText Generation"

[2] D. Demeter, "Analyzing the Role of Natural Language in NeuralSymbolic Models"

[3] P. Liang, M. Jordan, D. Klein, "Learning DependencyBased Compositional Semantics"

[4] T. Scialom, P. Dray, P. Gallinari, S. Lamprier, B. Piwowarski, J. Staiano, A. Wang, "QuestEval: Summarization Asks for Fact-based Evaluation"

[5] C. Liang, J. Berant, Q. Le, K. Forbus, N. Lao, "Neural Symbolic Machines: Learning Semantic Parsers on Freebase with Weak Supervision"

[6] M. Damonte, E. Monti, "One Semantic Parser to Parse Them All: Sequence to Sequence Multi-Task Learning on Semantic Parsing Datasets"

[7] C. Rebuffel, T. Scialom, L. Soulier, B. Piwowarski, S. Lamprier, J. Staiano, G. Scoutheeten, P. Gallinari, "Data-QuestEval: A Reference-less Metric for Data-to-Text Semantic Evaluation"

7 Appendix

7.1 Data Analysis-Addendum

The additional analysis performed on the datasets can be seen in Table 5.

7.2 Examples

7.2.1 Well generated triples

Example 1:

Original Text:
`The stylistic origin of sludge metal is hardcore punk.`

Actual Triples:
`Sludge_metal | stylisticOrigin | Hardcore_punk`

Description	Train	Dev	Test
Median tokens/text	18	18	20
1Q tokens/text	11	11	13
3Q token/text	26	26	29
——————-	–	–	–
Median Triples	3.0	3.0	3.0
# Triples - 1Q	2.0	2.0	2.0
# Triples - 3Q	4.0	4.0	4.0

Table 5: Addendum of Table 1

Predicted Triples from t5-base:
`Sludge_metal | stylisticOrigin | Hardcore_punk`

Example 2:

Original Text:
`Felipe Gozon is the key person at GMA New Media, a mass media company that offers products such as mobile apps. GMA New Media was founded on January 1, 2000 and is located inside GMA Network Center in the Philippines.`

Actual Triples:
`GMA_New_Media| foundingDate | 2000-01-01, GMA_New_Media | product | Mobile_Applications, GMA_New_Media | industry | Mass_Media, GMA_New_Media | keyPerson | Felipe_Gozon, GMA_New_Media | location | GMA_Network_Center, GMA_New_Media | location | Philippines`

Predicted Triples from t5-base:
`GMA_New_Media | foundingDate | 2000-01-01, GMA_New_Media | product | Mobile_Applications, GMA_New_Media | industry | Mass_Media, GMA_New_Media | keyPerson | Felipe_Gozon, GMA_New_Media | location | GMA_Network_Center, GMA_New_Media | location | Philippines`

7.2.2 Badly generated triples

Example 1:

Original Text:

```
English Without Tears is
89 minutes long.
```

Actual Triples:

```
English_Without_Tears |
runtime | 89.0
```

Predicted Triples from t5-base:

```
English_language
| language |
English_language,
English_language |
spokenIn | "Threes",
English_language
| language |
English_language
```

Example 2:

Original Text:

```
Brandon Carter, who was
born in England, is known
for Carter Constant and
his contributions to the
no-hair theorem.  His
doctoral advisor was
Dennis William Sciama.
```

Actual Triples:

```
Brandon_Carter | knownFor
| Carter_constant,
Brandon_Carter | knownFor
| No-hair_theorem,
Brandon_Carter |
doctoralAdvisor |
Dennis_William_Sciama,
Brandon_Carter |
birthPlace | England
```

Predicted Triples from t5-base:

```
Brandon_Carris_Carris
| doctoralAdvisor |
Dennis_William_Scama,
Brandon(_Carris)*18_Carreras
| birthPlace | England,
Brandon_Carris(_Carris)*95_Carr
```