Transformers for Textual Reasoning and Question Answering

Stanford CS224N Custom Project

Justin Wong
Department of Statistics
Stanford University
juswong@stanford.edu

Dominik Damjakob
Department of Statistics
Stanford University
damjakob@stanford.edu

Xinyu Hu
ICME
Stanford University
xhu17@stanford.edu

- Mentor: Drew Hudson

Abstract

Natural language models and systems have achieved great success in question answering tasks. However, much of the success is being measured on datasets such as SQuAD by Rajpurkar and Liang (2016) and RuleTakers by Clark et al. (2020) where questions simply require local phrase matching or shallow textual reasoning. As a result, the high performance transformers achieved on these tasks cannot demonstrate their ability to learn long-range relations and a holistic understanding of the text. We propose methods of reducing the attention mechanism of the transformer from a fully connected graph to one with sparser edge connections to see if it can yield improvements in performance for difficult reasoning tasks, generalizability, and learning efficiency.

1 Introduction

When it comes to modern natural language processing tasks, it is common practice to leverage transformer architectures. In particular, it has been found that its feed-forward repetition of dense network and attention layers are enough to surpass previous state of the art LSTM networks, even when combining LSTM with attention. Current state of the art methods often involve the pre-training of extremely large transformer models such as Bert or GPT and their variants on a huge corpus of public data and fine tuning these large models to the task desired.

The abundance of transformers in pushing performance metrics is of course due to their great ability to learn patterns and later apply them in a testing environment. However, this also tends to make them liable to learning spurious patterns and heuristics to perform well on data in unintended and potentially harmful ways. In particular, if the heuristics learned are not generalizable and the loss function and training data not expressive enough to discourage its usage, then the model will be a black box waiting to fail on the incorrect test case. It is therefore of utmost importance to learn models that are able to generalize well in the first place such that we have robust models to deploy in production environments.

To demonstrate that this is indeed a problem of concern, there are new results suggesting in (McCoy et al., 2019) that demonstrate that popular models, including Bert, often learn incorrect heuristics when trained on inference tasks. In particular, the original in-domain results demonstrate very respectable performance. However, when the authors shift the task to something out of domain (and contrary to the heuristic) the performance of these transformer models drop drastically. Furthermore, performance on a task which agrees with the learned heuristic performs much better than the in-domain task, which solidifies the claim. We expect that natural language understanding (NLU) systems are then susceptible to many other robustness issues, and many other problems with generalizability are discussed in (Johnson et al., 2017). For example, NLU systems are not robust to adversarial data nor shifts in dataset structure.

These shortcomings are not only a function of transformer architecture, but also demonstrate a need for more robust training data and losses that can better promote learning generalizable patterns. In this paper, we discuss methods of improving the transformer attention mechanism and how using a synthetic dataset to test the systematic generalization and inductive reasoning capabilities of our models is useful to robust learning.

2 Related work

There are many impressive results for various question answering tasks in NLP. However, previous benchmarks include the Stanford Question Answering Dataset (SQuAD) by Rajpurkar and Liang (2016), the Stanford Natural Language Inference (SNLI) corpus by (Bowman et al., 2015), the RuleTakers dataset by Clark et al. (2020) and more which mostly focused on factual questions and sentence understanding. However, these tasks all fall under a similar problem where the simple require either local phrase matching or shallow textual reasoning in order to succeed. Hence, the tasks they require models to learn can often be simplified to simpler heuristics that may not generalize well. For example, SQuAD can be reduced to a task of matching the context of the question to the context of the passage and finding the corresponding phrase and RuleTakers can be reduced to a task of matching the context of the question and doing a sentiment analysis. In other words, the model may be learning to relying on dataset-specific artifacts instead of learning robust and generalizeable ideas. This differs from our approach to generate a synthetic benchmark for systematic generalization and inductive reasoning.

The Compositional Language Understanding and Text-based Relational Reasoning (CLUTRR) suite is introduced by Sinha and Hamilton (2018) and highlights many of the baselines for this project. The new datasets introduces the task of learning personal relations under various perturbations. These involve difficult learning scenarios including inference under various forms and amounts of noise, performing different inference length tasks, and performing memory tasks all of which must be done in and out of domain. The authors also demonstrate that current NLU models perform poorly compared to structured graphical models. With the experimental setup introduced in this paper, it was observed that the Graph Attention Network (GAT) model outweighed all NLU models. Within the text-based models, BERT-LSTM is the consistent top-performer. We first use their evaluation methods as a baseline to build our own custom models to compare. We also use their data generation methods as a baseline and extend it to produce other tasks that we wish to evaluate on.

Shanthamallu et al. (2020) discuss methods of modifying traditional Graph Attention Networks (GATs) to be more robust to noise in data. The authors attempt to achieve this by implementing regularization terms that penalizes uniformity and encourages sparsity in the graph attentions. This method demonstrates notable improvements in accuracy when evaluated on a modified attention network that selects and randomizes attention for some number of nodes. This method is a good first step towards robustness and we adopt an approach inspired by the authors to encourage a slightly different attention behaviour in our own model.

Hudson and Zitnick (2021) discuss the shortcomings of canonical feed-forward architecture of computer vision models from lower level details to higher level features and highlights the difference from human processing which also involves a reverse flow of information which consolidates to form richer human interpretations of images as compared to neural network models. In order to include this bidirectional flow of information in neural network models as well, the authors propose a bipartite attention structure between image features and latent variables that allows for interplay of the two to inform the attention weights. While this is applied to image processing, a similar analog can be made for textual processing since the reverse flow of information can use surrounding context and selective attention to better inform reasoning tasks. We adopt a similar methodology as an experimental model architecture.

3 Approach

3.1 Baselines

In our research, we utilize pretrained transformer models, namely BERT (Devlin et al., 2018), to measure baseline performance. We evaluate our models using the CLUTRR codebase and test for generalizability by applying our models to noisy CLUTRR and other variants. While (Devlin et al.,

2018) only train the decoder on top of their BERT model, we will also use a fine-tuned BERT models as a more competitive baseline.

3.2 Approach

In our approach, we fine-tune BERT models to the reasoning task and enhance them by requiring sparsity for the Graph Network implied by the transformer encoding. This can be achieved by including regularization terms to the loss function or pruning on pre-trained models and specialized version of multiheaded attention. Apart from effect of sparsity, we are also curious about the combination result of transformers and pretrained models.

To start with, regularization is a good way of encouraging attention sparsity, as introduced in (Shanthamallu et al., 2020). Specifically, they add penalty terms to the loss function. Storing the attention coefficients to K heads in the attention adjacency matrix $A^k \in \mathbb{R}^{N\times N}, k \in \overline{1,K}$, they introduces two regularizing terms

$$L_{excl} = \frac{1}{NK}\sum_{k=1}^{K}\sum_{j=1}^{N}\sum_{i=1}^{N}|A_{ij}^k|$$

which prevents nodes from becoming discordantly influential and

$$L_{nonunif} = -\frac{1}{K}\sum_{k=1}^{K}\frac{1}{N}\sum_{i=1}^{N}||A_i^K||_0 - N$$

which encourages edge sparsity. However, these two regularizations are insufficient for our application as the l_1 regularization is diminished by the specifications of the attention function which requires that $\sum_{j=1}^{N}|A_{ij}| = 1$. Also, the introduction of an l_0 regularization would result in a stark discontinuity of the loss function's gradient. Therefore, we propose the regularization

$$L_{distributional} = \frac{1}{K}\sum_{k=1}^{K}||\frac{1}{N}\sum_{i=1}^{N}A_i^K||_\mu \tag{1}$$

where we choose a small parameter for μ, namely $\mu = 0.1$ in our initial experiments.

Here, we consider specialized version of multiheaded attention, where the same node sparsity and importances between each of the different heads are enforced. This brings the further benefit of applying its sparsity constraint over all attention heads at once, thus requiring that the overall effect of the attention head modulates a sparse graph structure. Thus, we use the regularization

$$L_{distributional} = ||\frac{1}{K}\sum_{k=1}^{K}\frac{1}{N}\sum_{i=1}^{N}A_i^K||_\mu \tag{2}$$

which provides the further benefit of applying its sparsity constraint over all attention heads at once, thus requiring that the overall effect of the attention head modulates a sparse graph structure.

We use these sparsity restrictions in two ways. In the first, we generate a transformer network that applies the penalization term specified in Equation 2 and stick it on top of a standard BERT model. This allows the BERT model to incorporate rich relationships between the words while the additional transformer layers can model the graph network implied by the logical relationships. In a second variant, we prune the BERT model directly by fine-tuning it under the distributional penalization scheme from Equation 2. This allows us to incorporate the penalization function more elegantly and the words will be focused on their logical counterparts from the start.

4 Experiments

4.1 Data

The main dataset that we will use is the CLUTRR suite which is composed of a set of semi-synthetic passages focusing on familial relationship Sinha and Hamilton (2018). The authors generate this

dataset by first gathering fundamental data on named entities and sample phrases from Amazon Mechanical Turkers. This simply gives the text corpus more flavor and models realistic text that may be found in books or websites such that the model still needs to resolve canonical tasks such as co-reference resolution, dependency parsing and named entity recognition on top of learning a relationship tree. Then, logical rules governing relationship composition must be added and we can begin generating data.

Generation is done by first initializing a single person node. We can then add other nodes that relate directly to the first node to begin a relationship tree. More nodes are recursively added until it reaches a desired size after which we can apply transformations. In particular, we construct the graph downwards and by using compositional rules and inverse compositional rules we can perturb nodes to construct the upper levels and to draw edges between all relations that can be named to form the final relationship graph (note that this will no longer be a tree). Then in order to generate a (corpus, query) pair, we sample a edge in the graph such that there exists a different path between the nodes and construct a story using the sample phrases data and fundamental data and query the relation between the nodes connected by the sampled edge. Of course the edge is the final target relation.

The implementation baseline is provided in https://github.com/facebookresearch/clutrr (2019) and does generation specifically on familial relationships. The authors also provide some simple baseline data generated from the model which is very simple in nature and only tests simpler reasoning capabilities. In order to challenge the baseline models and modified models we propose more, we require more challenging data.

First, CLUTRR is written specifically for familial relations and the generation functions are written particularly for this generation task. In order to see if the models generalize to other relationship types and not just a unique fit to familial data, we extend CLUTRR to generate some other relationship type which we choose to be workplace data. We build off of the familial generation structure and add additional workplace data that is generated as a full tree alongside family such that we can now add workplace relation queries as well to our training and testing data.

Next, we also wish to challenge our models to generalize to perform longer inference tasks as well as do robust reasoning on noisy text. The CLUTRR API allows us to specify different generation procedures of the form $x.y$. The naming convention here is x denotes a task ID and y denotes the inference length of queries in the dataset. In order to test on longer inference tasks, we want to vary the length of alternative path required to traverse the edge sampled during generation. Then for any task, we generate a training set for smaller inference lengths (i.e. $y = 2, 3, 4$) and tested on inference lengths that reach higher (i.e. $= 5, 6$). For robust reasoning, note that once the relationship graph is fully built, we can add additional edges for non-relationship information. This allows us to add non-essential information to the tasks as a form of noise that will require models to focus on only necessary information. We can generate some different forms of noise: no additional noise ($x = 1$), supporting noise ($x = 2$) where the edges added follow to alternative path between the queried nodes, irrelevant noise ($x = 3$) where the edges added connect to one of the queried nodes, and disconnected noise ($x = 4$) where the edges added no not connect to either of the queried nodes. Finally, we have tasks $x > 5$ are such that the target is in the text corpus and tasks $x < 6$ are such that the target is not in the text corpus.

4.2 Experimental details

We generated and tested five models built on top of the https://github.com/koustuvsinha/clutrr baselines (2019) CLUTRR baselines evaluation code. These are a standard BERT model with added Feed-Forward architecture, BERT with an added transformer structure and 3 versions of BERT with added regularized transformers. These 3 models use $\mu = 0.1$ and have different λ parameters by which we multiply the regularization term when adding it to the loss function, specifically $\lambda = 0.01, 0.1, 1$. For all our regularization models we use the penalization terms defined in Equation 2 to enforce sparsity across the attention heads.

For simplicity, we initially tested our models on the pre-generated CLUTRR data sets. In particular, we train our models on the pre-generated CLUTRR data that is immediately available for download https://drive.google.com/file/d/1SEq$_e$1$IVCDDzsBIBhoUQ5pOVH5kxRoZF/view$ (2019). The training data includes text passages generated using stories involving family relations (train 2.2) and supporting facts and testing data involved a slightly different structure of family

relations with irrelevant facts mixed in (test1.3, test 2.3, test 3.3, test 4.3). The task queries are given by the names of two individuals ($Person_1$, $Person_2$) in the text and the model is asked to output the word relating $Person_2$ to $Person_1$ (i.e. $Person_2$ is the son of $Person_1$).

4.3 Results

4.3.1 Precompiled Data Sets

Our results are summarized below. Table 1 shows accuracy statistics for different combinations of transformer and BERT models on the training and test data precompiled by Sinha and Hamilton (2018). What we can see it that fine-tuned BERT models generate much higher accuracy statistics than simple pre-trained BERT models with an added transformer structure. This holds for test tasks that are the same as the training tasks, but also for other test tasks such as tasks 3.3 and 4.3. Further, this is also true for our baseline models, as the fine-tuned BERT models outperform simple pretrained BERT by a lot. We can also note that the BERT models that only fine-tune their attention layers tend to achieve better accuracy statistics, and that our regularized models are often able to outperform their non-regularized variants. Because we can observe a stark difference between added-transformer and pruned BERT models, we will focus on the latter in the next result sections.

Table 1: **Model Average Accuracy Table**
Test accuracy over 5 data sets from (https://github.com/koustuvsinha/clutrr baselines, 2019) with different levels of noise relations. Training sets are 1.2 and 1.3

	1.2	1.3	2.3	3.3	4.3
BERT	0.292	0.393	0.541	0.430	0.516
BERT with transformer	0.047	0.286	0.361	0.211	0.133
BERT with regularized transformer, $\lambda = 1$	0.109	0.250	0.117	0.188	0.172
BERT with regularized transformer, $\lambda = 0.1$	0.417	0.273	0.339	0.320	0.320
BERT with regularized transformer, $\lambda = 0.01$	0.161	0.281	0.219	0.211	0.344
BERT fine-tune	0.740	0.771	0.731	0.773	0.766
BERT fine-tune attention	0.693	0.826	0.836	0.805	0.836
BERT fine-tune reg., $\lambda = 1$	0.677	0.773	0.541	0.695	0.727
BERT fine-tune attention reg., $\lambda = 1$	0.724	0.865	0.859	0.812	0.805

4.3.2 Workplace Data

To test whether for logical reasoning capabilities across data sets, we combined the workplace and family relation data sets for both training and testing sets. Accuracy statistics can be found in Table 2. As all accuracy statistics except for tasks 4.2 and 5.2, for which we get accuracy rates around 0.8, are close to 1, we can conclude that the models are not bound to a single task. Further, we can not that our restricted model also works with other versions of the penalization parameter λ and that an increase of the parameter does not lead to a sharp performance decrease. This means that our model can also function with stricter attention restrictions.

Table 2: **Model Average Accuracy Table, Robust across environments**
Test accuracy using both the family-relationship and workplace data sets. The models were trained on tasks 1.2, 3.2 and 6.2

	1.2	2.2	3.2	4.2	5.2	6.2	7.2
BERT	0.994	0.597	0.872	0.600	0.508	1.00	0.993
BERT fine-tune	1.000	0.825	1.000	0.730	0.696	1.00	1.000
BERT fine-tune attention	1.000	0.908	1.000	0.857	0.826	1.00	1.000
BERT fine-tune reg., $\lambda = 0.1$	1.000	0.720	0.872	0.736	0.495	1.00	1.000
BERT fine-tune attention reg., $\lambda = 0.1$	1.000	0.882	1.000	0.865	0.842	1.00	1.000
BERT fine-tune attention reg., $\lambda = 0.5$	1.000	0.872	0.986	0.886	0.766	1.00	1.000
BERT fine-tune attention reg., $\lambda = 1$	1.000	0.892	1.000	0.891	0.769	1.00	1.000

4.3.3 Longer relationships

Logical reasoning models need to be able to reason over longer logical chains. Therefore, Table 3 provides accuracy statistics for longer test sets for a model trained on tasks 1.2, 1.3 and 1.4. The statistics show that more robust models are better able to correctly predict long relationship solution, while the pre-trained BERT and the simple fine-tuned BERT models are inadequate for such tasks. Yet, the relatively low accuracy rates for longer tasks, such as 1.6, even for restricted models indicates that all models have problems with long reasoning tasks given the training sets.

Table 3: **Model Average Accuracy Table, Long Reasoning capabilities**
Test accuracy for logical chains of different lengths. Training sets are 1.2, 1.3 and 1.4

	1.2	1.3	1.4	1.5	1.6
BERT	0.604	0.251	0.220	0.213	0.148
BERT fine-tune	0.891	0.765	0.469	0.380	0.349
BERT fine-tune attention	0.823	0.765	0.517	0.425	0.303
BERT fine-tune reg., $\lambda = 0.1$	0.891	0.694	0.491	0.378	0.279
BERT fine-tune attention reg., $\lambda = 0.1$	0.953	0.721	0.486	0.408	0.248
BERT fine-tune attention reg., $\lambda = 0.5$	0.969	0.793	0.480	0.404	0.271
BERT fine-tune attention reg., $\lambda = 1$	0.969	0.800	0.469	0.404	0.295

To check whether the models only generated worse accuracy statistics for longer reasoning tasks due to inadequate training data, Table 4 provides experiments using longer training data sets. Due to the expanded training data, we incorporated test data of reasoning length up to 10. As the table shows, expanding the training data length raises the predictive capacity for all models, and especially the fine-tuned models are now able to almost always make correct inferences for test sets of up to length 10. However, for longer tasks we can observe a sharp performance reduction, though it is less pronounced for more robust models. In general, especially the regularized models with $\lambda = 0.1$ and $\lambda = 0.5$ are able to achieve a relatively good performance when compared to the other models.

Table 4: **Model Average Accuracy Table, Long Reasoning with longer training data**
Test accuracy over 5 data sets from (https://github.com/koustuvsinha/clutrr baselines, 2019) with different levels of noise relations. Training sets are 1.2, 1.3 and 1.4, 1.5 and 1.6

	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	1.10
BERT	0.302	0.257	0.151	0.448	0.538	0.146	0.106	0.154	0.193
BERT fine-tune	0.984	0.691	0.518	0.990	0.975	0.243	0.362	0.307	0.285
BERT fine-tune attention	0.984	0.760	0.498	0.990	0.950	0.256	0.263	0.378	0.395
BERT fine-tune reg., $\lambda = 0.1$	0.984	0.699	0.477	0.979	0.975	0.272	0.276	0.267	0.307
BERT fine-tune attention reg., $\lambda = 0.1$	0.984	0.744	0.425	0.979	0.954	0.313	0.352	0.350	0.391
BERT fine-tune attention reg., $\lambda = 0.5$	0.984	0.807	0.451	0.990	0.975	0.276	0.387	0.387	0.356
BERT fine-tune attention reg., $\lambda = 1$	0.984	0.752	0.518	0.990	0.975	0.272	0.273	0.331	0.329

5 Analysis

To investigate the functional performance of our regularized models, we inspect the attention layers of selected models in this section. The purpose of our Distributional regularization scheme is that it aims to restrict the attention connections in this layer. As the attention network of a transformer can be viewed as a Graph Neural Network, where the tokens a token attends to denote its connected nodes in the Graph Network, our penalty function can generate a sparser graph structure. This is especially true as the regularization works across the different attention head in a single layer and thus restricts the total graph connections symbolized by the attention network. Due to our choice of the norm μ, which we have set to $\mu = 0.1$, the layer is incentivised to restrict its attention to a small number of tokens, as the penalty term discourages a more uniform distribution.

For the inspection, we choose fine-tuned BERT models trained on the pre-generated data set; in particular, we use our model without a penalty term, as well as the models with $\lambda = 0.1$ and $\lambda = 1$ to inspect for the effects of different strengths of the regularization. Figures 1 and 2 depict attention layers for 2 sample inputs from the original data sets for the three input models. For the input

sentences, the numbers denote a token used in place of a person (for example, the start of the story in Figure 1 reads "Person 0 and her daughter person 1 went shopping together").

As expected, Figure 1 shows that indeed with an increasing degree of regularization (with the figures progressing from $\lambda = 0$ to the left to $\lambda = 1$ to the right) the attention graph becomes sparser, whilst it retains its ability to capture the key information and relationships. Specifically, especially up to $\lambda = 1$ for this sentence, the first two versions capture the connection between the name represented by the 1 token and the relationship word daughter. Further, we can observe that such crucial relationships may be less influential for more extreme regularization term if some noise component overshadows the relative relationship, as is the case for $\lambda = 1$.

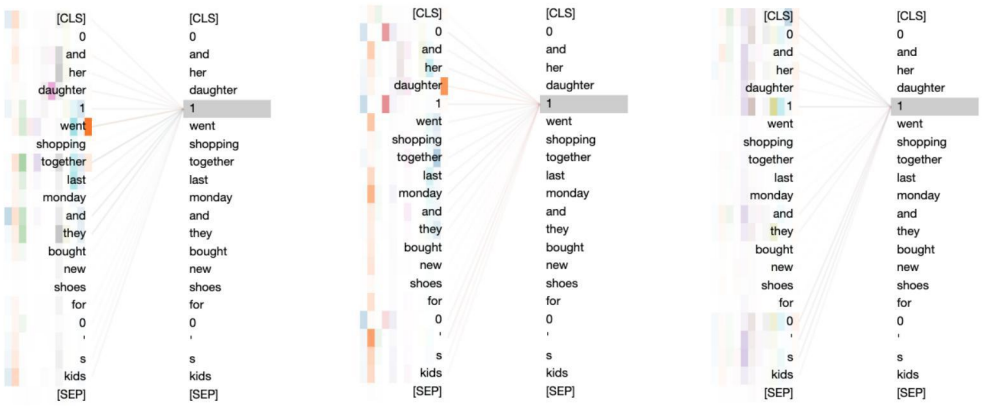

Figure 1: **Attention visualization of different degrees of penalties** left to right: $\lambda = 0, 0.1, 1$

Similarly, Figure 2 visualizes an attention layer for another sentence from the data base. Again, we can see that the attention outputs are indeed less dispersed and thus symbolize a more restricted graph network. For the token character 1, with increasing λ values it attends more strongly to 'mother', the word that describes its relationship with character 2. Thus, in this case does not only help in eliminating - or pruning - non-necessary attention layers, but can also help in identifying correct relationships, even for large λ values.

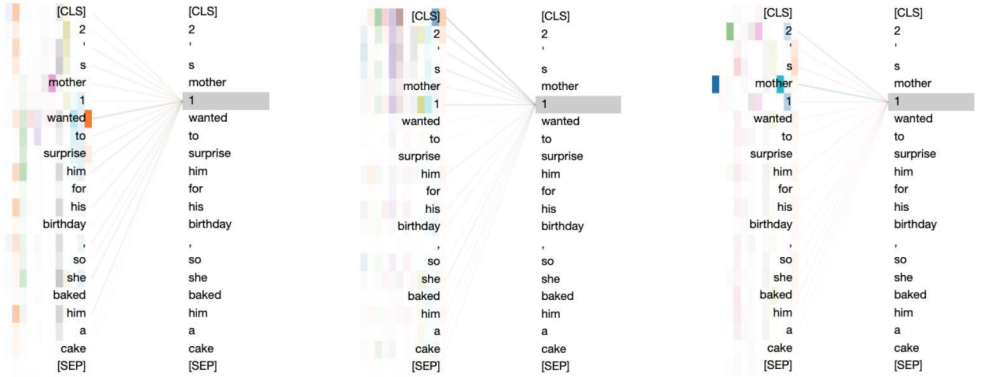

Figure 2: **Attention visualization of different degrees of penalties** left to right: $\lambda = 0, 0.1, 1$

6 Conclusion

In this research paper, we investigated the possibility to achieve better reasoning capabilities for Natural Language Understating in a reasoning task by using a sparser and more restricted network. In particular, we performed our analysis on the CLUTTR data set, which contains statements and

questions about family relationships that require long-term reasoning capabilities. As inferences across multiple connections can be difficult, this was established to be a challenging task according to prior literature.

In our work, we extended the CLUTRR data set to a second task by activating its possibility to use relationships fro a work environment. This introduces the additional requirement that models need to apply reasoning across different environments and thus requires greater generalizability. We have further expanded across the state-of-the-art performance established by Sinha and Hamilton (2018) by using regularized version of fine-tuned BERT models. Our performance gains are particularly robust to the introduction of additional noise variables, such as non-necessary connections. However, while the regularized BERT models also increase model performance for long-relation tasks, the increase in predictive accuracy dissipates for longer connections. This shows that despite the performance improvements, the models are still unable to generally model the ability to conduct logical reasoning. This provides an important direction for future research, as models with the ability to perform such long-distance reasoning can lead to more reliable NLP and NLU models that take the syntactic content of their texts into account.

Further, we have successfully introduced the distributional regularization attention model, which is the first attention model to directly restrict graph connections across attention heads by a penalization method. Given the success in traditional Machine Learning, but also its recent re-emergence in Deep Learning, regularization techniques can play an important part in the future. As shown, our regularization method can restrict the attention graph to a more sparse structure, with the sparsity depending on the strength of the regularization. Further, this method allows to prune pre-trained models by fine-tuning them under a regularization scheme. This allows to employ less computation resources when designing a sparse model, and to design the model for a relatively small training set, as in this paper. Yet, more experiments are needed to reveal the full potential of the distributional regularization scheme. More experiments should be conducted to identify optimal parameter values for μ and λ, and their effect on other graph-based attention tasks should be researched.

Lastly, the researchers have also investigated the possibility of introducing the new Simplex Attention model to the NLP literature to design sparse graphs more explicitly. By defining a separate, more sparse set used to compute key-value pairs for the attention layer, the Simplex function can design its own sparse transformer models and does not require additional regularization schemes. However, at this stage the Simplex results were not yet satisfactory, such that it was omitted from the result section.

References

Bowman, S. R., Angeli, G., Potts, C., and Manning, C. D. (2015). A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*.

Clark, P., Tafjord, O., and Richardson, K. (2020). Transformers as soft reasoners over language. *arXiv preprint arXiv:2002.05867*.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

https://drive.google.com/file/d/1SEq$_e$1$IVCDDzsBIBhoUQ5pOVH5kxRoZF/view$(2019).$Clutrrpre-generateddownload$.

https://github.com/facebookresearch/clutrr (2019). Clutrr.

https://github.com/koustuvsinha/clutrr baselines (2019). Clutrr.

Hudson, D. A. and Zitnick, L. C. (2021). Generative adversarial transformers. *arXiv preprint arXiv:2103.01209*.

Johnson, J., Hariharan, B., Van Der Maaten, L., Fei-Fei, L., Lawrence Zitnick, C., and Girshick, R. (2017). Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2901–2910.

McCoy, R. T., Pavlick, E., and Linzen, T. (2019). Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. *arXiv preprint arXiv:1902.01007*.

Rajpurkar, P., Z. J. L. K. and Liang, P. (2016). Squad: 100,000+ questions for machine comprehension of text. In *arXiv.1606.05250*.

Shanthamallu, U. S., Thiagarajan, J. J., and Spanias, A. (2020). A regularized attention mechanism for graph attention networks. *arXiv preprint arXiv:1811.00181*.

Sinha, K., S. S. D. J. P. J. and Hamilton, W. L. (2018). Clutrr: A diagnostic benchmark for inductive reasoning from text. In *arXiv.1908.06177*.