Exploring Semantics in Pretrained Language Model Attention

Frédéric Charpentier, Jairo Cugliari, Adrien Guille

Exploring Semantics in Pretrained Language Model Attention

Frédéric Charpentier[1,2] and **Jairo Cugliari**[1] and **Adrien Guille**[1]
[1]Université Lyon 2, Laboratoire ERIC, Lyon, France
[2]Cabot Financial France
{frederic.charpentier, jairo.cugliari, adrien.guille}@univ-lyon2.fr

Abstract

Abstract Meaning Representations (AMRs) encode the semantics of sentences in the form of graphs. Vertices represent instances of concepts, and labeled edges represent semantic relations between those instances. Language models (LMs) operate by computing weights of edges of per layer complete graphs whose vertices are words in a sentence or a whole paragraph. In this work, we investigate the ability of the attention heads of two LMs, RoBERTa and GPT-2, to detect the semantic relations encoded in an AMR. This is an attempt to show semantic capabilities of those models without finetuning. To do so, we apply both unsupervised and supervised learning techniques.

1 Introduction

An AMR graph, as specified by Banarescu et al. (2013), is a representation of the meaning of a sentence in the form of a directed acyclic graph, involving concepts from neo-Davidsonian semantics (Davidson, 1969). A number of datasets of sentences and their corresponding hand-crafted AMRs have been published, and various techniques have been developed to automatically build AMR graphs from sentences in natural language. These include graph based approaches, which directly predict nodes and edges from the sentences, (Flanigan et al., 2014, Zhang et al., 2019), and algorithms based on transition systems (Nivre, 2008), inspired by dependency parsing algorithms (CAMR: Wang et al., 2015, AMR-Eager: Damonte et al., 2017). The most recent solutions combine an encoder-decoder pair of a transformer network (Vaswani et al., 2017) to adapt it to the task of transition-based AMR parsing, as StructBART does (Zhou et al., 2021).

AMR graphs abstract away meaning from syntactic representations. This is evidenced by the fact that one AMR graph can encode several differently worded sentences, even with different syntaxes. (Banarescu et al., 2013)

Transformer-based language models, introduced by Vaswani et al. (2017), have shown remarkable performance in solving many problems related to automatic natural language processing, but the interpretability of their computations is still subject to active research: Clark et al. (2019) studied the ability of certain attention heads in the BERT network (Devlin et al., 2019) to classify several syntactic relations between words and to resolve coreference, without finetuning BERT for any specific task. Luo (2021) studied how constituency grammar is captured by different attention heads in BERT. We complement their work and explore the ability of attention heads to classify semantic relations between two words as described by the edge type between two vertices of an AMR.

We study a representative bidirectional pretrained language model, without finetuning: RoBERTa (Liu et al., 2019), and compare it to GPT-2 (Radford et al., 2019), a pretrained conditional model, using both unsupervised and supervised techniques. Our study reveals a striking correlation of these networks' attention heads with semantics. We observed that RoBERTa showed conspicuously better results than GPT-2, probably because of the bidirectional nature of the former. To reproduce our experiments, we made our code publicly available.[1]

2 Dataset Design and Experimental Setup

In a nutshell, an AMR encodes in a rooted directed acyclic graph **who** is doing **what** to **whom**, **where**, **when** and **how**, in a manner that abstracts away semantics from syntax. In particular, a single AMR graph can encode several syntactically different sentences, like "the bears invaded Sicily" (a whole clause), "the bears' invasion of Sicily" (a noun

[1]https://anonymous.4open.science/r/sem_LM_att-322F/

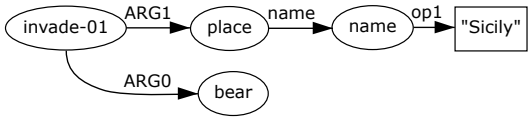


Figure 1: AMR for several wordings: "The bears invaded Sicily", "The bears' invasion of Sicily", "The invasion of Sicily by the bears" and "The invasion of the bears in Sicily".

phrase), "the invasion of Sicily by the bears", or "the invasion of the bears in Sicily". See Figure 1. In so doing, an AMR encodes instances of concepts as vertices in the graph, using PropBank framesets (Palmer et al., 2005) wherever possible[2]. Relations between instances of concepts are encoded as directed labeled edges in the graph. Those relations can be the frame arguments, following PropBank conventions (`ARG0`, `ARG1`,...), or other general semantic relations (`time`, `cause`, `location`, *etc.*).

Blodgett and Schneider (2021) published a dataset of automatic alignments between AMRs and the corresponding English sentences in the LDC2020T02 dataset (Knight et al., 2020), which comprises 59,255 sentences. We took advantage of those alignments and built a **dataset of edges** to test the capability of an LM's attention mechanism to retrieve the semantic relation encoded in a edge from the two connected vertices.

In their work, Blodgett and Schneider labeled their alignments across several categories: • **subgraph alignments**, a mutually exclusive alignment between consecutive spans of words and subgraphs of the AMR, • **duplicate subgraph alignments**, to account for elliptical construction, where several identical subgraphs in the AMR are mapped to the same word span, • **relation alignments**, providing alignments between a span and a single relation, (an arc in the AMR), such as "when" → `:time`, and • **reentrancy alignments**, accounting for reentrancy, (the fact that an AMR node may have multiple incoming edge). Reentrancy alignments provide alignments between reentrant arcs and a word span that triggers the reentrancy. (Pronouns, control verb, etc.) We selected the "subgraph alignments", sorting them to keep only those alignments involving a single word in the sentence and a single-vertex subgraph. Next, we had the sentences processed by the tokenizers of two pretrained language models with 12 layers and 12 heads per layer : • **RoBERTa**, a bidirectional encoder (Liu et al., 2019), and • **GPT-2**, a conditional decoder (Radford et al., 2019). To deal with the case of words split across several tokens, we aligned the sequences of words with the sequences of tokens, keeping only alignments involving a single word aligned with a single token.

[2]For example, the noun "invasion" and the verb "invaded" are both encoded using the PropBank frame `invade-01`.

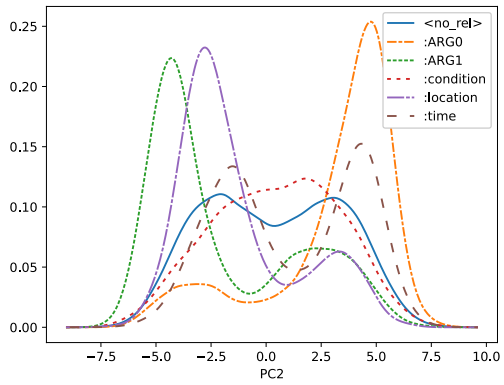


Figure 2: Distributions of six relations on the RoBERTa encoder, projected on PC2.

We thus created a dataset of pairs of tokens aligned with vertices of AMR graphs, linked by a semantic relation. To assess the ability to detect the absence of a semantic relation, we included random pairs of tokens corresponding to vertices in the AMRs with no edge between them, to create a category "<no_rel>". We then ran the transformers with all sentences as input and computed their representations. For each pair of words, there are possibly two attention directions to be computed: attention from one word to the other, or conversely. We call them ST and TS, as they represent attention from the source to the target or from the target to the source, where "target" and "source" denote the direction of the edge in the AMR graph. Each of those two attentions is represented by 144 scalars, as there are 12 layers, and 12 attention heads per layer.

In a transformer, the attention weight from a source token Q to a target token K is obtained by taking two affine transformations of the embeddings of Q and K, computing the inner product of those, and taking the softmax of that product with respect to all other target tokens. The features we use throughout this study are actually those inner products, before application of the softmax.

2.1 Illustration

As an example, let us consider the sentence "Establishing Models in Industrial Innovation". Its

AMR displays an edge ":ARG1" between the node "innovate-01" and the node "Industry". The alignments indicate a subgraph-alignment between the node "innovate-01" and the word "innovation", and another alignment between the node "industry" and the word "industrial". Both words "industrial" and "innovation" correspond to a token in the transformer model, therefore the edge labeled ":ARG1" could be kept in the dataset. The corresponding entry consists of the 144 features of attention from the token "industrial" to the token "innovation", and the 144 features from the token "innovation" to the token "innovation". The label is ARG1.

3 Unsupervised Analysis: PCA

The first step of our study was to apply a simple dimension reduction technique to the dataset. We chose to compute a principal component analysis on the inner product dataset. For the dataset computed with RoBERTa, we found that keeping 4 principal components enabled us to explain about 52% of the total variance.

We filtered the dataset by relation, and computed kernel density estimations of the distribution of different relations, and looked for dissimilarities between those. To do so, we selected a few relations to be plotted overlaid : (<no_rel>, ARG0, ARG1, condition, location, time, ARG2, quant, polarity, mod, and poss). We found that the first six relations could easily be distinguished by examining only the projection on the second principal component, as their distributions seem very different, although somewhat overlapped (See Figure 2).

The most conspicuous separation is between ARG0 and ARG1. However, no pair of relations presented completely distinct distributions. The separation of the relations quant, mod and poss is less obvious, and can best be seen on the projection on the fourth principal component. As for the pair of relations (ARG1, ARG2), they can hardly be distinguished. (Plots can be seen on appendix A.1)

For the dataset computed with the decoder GPT-2, we found that keeping 4 principal components enabled us to explain 70% of the total variance. (18% more than with RoBERTa). In spite of this difference, we found that GPT-2 was less effective than RoBERTa in distinguishing relations. As an illustration, in Figure 3, we plotted the distributions of the three easiest to distinguish relations for RoBERTa and GPT-2 on the most distinctive axis

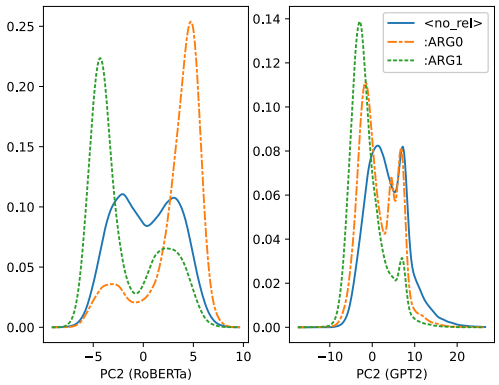


Figure 3: Distributions of relations <no_rel>, ARG0 and ARG1 on RoBERTa and GPT-2, showing the better separation of RoBERTa.

PC2. (See appendix A.2 for more GPT-2 plots.)

This first step tends to show that it is possible to use the attention heads of a transformer network to observe different distributions for pairs of different semantic relations. This behavior of a transformer is more prevalent for a bidirectional encoder (like RoBERTa) than for a conditional decoder (like GPT-2).

4 Supervised Analysis: Logistic Classifier

On the strength of these results, we trained a logistic model to classify the semantic relations of our datasets. To do so, we modified the datasets in the following way :

1. For RoBERTa, we included both ST and TS attention features, thus amounting to 288 features per sample.

2. We left out from the dataset some relations which we deemed irrelevant for semantics : snt-n, used to point to numbered independent clauses in a sentence, op-n, used for coordination with conjunctions like "and", "or", or commas, or for numbering the parts of a composite named entity, and polarity, whose target is almost always the constant "negative", and not an instance of a concept.[3]

3. Since the dataset is highly unbalanced, we grouped every relation with fewer than 1000 samples under the general category <other>, gathering 2.1% of our data.

[3] polarity is used to signal that a sentence is negative.

4.1 RoBERTa Language Model

Eventually, we obtained for the RoBERTa dataset 375,335 samples divided into 18 semantic relations to be classified. We then trained a Logistic classifier, using class weighting to compensate for the imbalance. The global balanced accuracy on test data is 0.62.[4] Detailed results are shown in the left column of Table 1.

Classes ARG0, ARG1, time, mod, quant exhibit the best F1 scores, with respectively 0.74, 0.63, 0.63, 0.63 and 0.60. Besides <other>, ARG3, ARG4 and topic are the classes showing the worst F1 scores (0.09, 0.13 and 0.17). This is probably because ARG3 and ARG4 are used in some PropBank frames to describe a role where other AMR relations could arguably have been used. (price, instrument, reason, location). Relations topic and condition also exhibit a poor F1 score of 0.17 and 0.20. Interestingly enough, a careful scrutiny of the confusion matrix shows that many false positives for topic are confusions with ARG1, mod and <no_rel>, entailing a poor precision for this relation. The recall is otherwise good. This is also the case for condition. (See Appendix C.1 for the confusion matrix.)

4.2 GPT-2 Language Model

For the case of GPT-2, the very nature of a decoder does not allow attention to be computed in both directions, but only from a subsequent token to its predecessors. Therefore, we could only take advantage of 144 features. The global balanced accuracy is 0.44, and individual F1 scores are reported in Table 1. They are much poorer than the results obtained with RoBERTa, with which we used the full number of 288 features. We made the hypothesis that the reduced number of features due to causal self-attention is detrimental to a good detection of the semantics. To confirm this idea, we modified the implementation[5] to output the full inner products tensors used in computing attention before masking, without altering the network's operation. We trained another logistic classifier on this new dataset, and reported the results in the right column of Table 1. Every single F1 score is better than the scores obtained on the plain GPT-2, and the global balanced accuracy amounts to 0.56, a gain of more than 11 points.

Relation	Freq	RoBERTa	GPT2	GPT2 aug.
ARG0	16%	0.74	0.60	0.69
ARG1	33%	0.63	0.34	0.55
time	3%	0.63	0.29	0.54
mod	12%	0.63	0.44	0.57
quant	1%	0.60	0.40	0.56
<no_rel>	17%	0.59	0.44	0.49
degree	1%	0.54	0.35	0.52
poss	1%	0.47	0.20	0.32
location	1%	0.45	0.22	0.37
part	0.4%	0.37	0.11	0.24
manner	1%	0.36	0.16	0.28
ARG2	8%	0.33	0.20	0.29
purpose	1%	0.31	0.18	0.23
condition	1%	0.20	0.14	0.16
<other>	2%	0.18	0.08	0.15
topic	1%	0.17	0.11	0.14
ARG4	0.5%	0.13	0.07	0.10
ARG3	1%	0.09	0.08	0.11

Table 1: F1 scores per class of the Logistic classifier trained on the three datasets: RoBERTa, GPT2 and GPT2 augmented.

4.3 Influence of the Heads on the Results

The nature of a Logistic classifier allows us to interpret the contribution of the different heads to the detection of a relation by analyzing the coefficients of the classifier. Specifically, we can determine if an increment in the response of a particular head increases or decreases the ratio of probabilities of two relations.The following study was conducted on RoBERTa, we left GPT-2 aside. First, we analyzed the ratio of all probabilities with respect to the probability of <no_rel>. For that purpose, we computed the differences between the coefficients of all linear predictors and the coefficients of the linear predictor for <no_rel>. We noticed that for head 3 in layer 4, (head H3L4), as well as heads H1L6 and H2L3 of the TS product, all those differences were negative. This means that any positive shift in the inner product computed by one of those heads induces an increase of every ratio $\frac{\mathbb{P}[y=\texttt{<no\_rel>}]}{\mathbb{P}[y=y_k]}$, for all $y_k \neq$ <no_rel>. Conversely, we noticed that a positive shift in heads H5L8 or H3L9 (both for ST attention) induced an increase of the inverse ratio. We can conclude that those heads are specialized in determining a semantic relation, or absence thereof.

We further analyzed the contribution of every head to the probability ratio of any pair of relations: for each possible pair of relations, we recorded the k most contributional heads to the direct probability ratio, as well as the top k heads for the inverse

[4] In comparison, random forests and MLP classifiers have slightly poorer precision.

[5] We used minGPT, (https://github.com/karpathy/minGPT), which we deemed the easiest to modify, while providing a complete implementation.

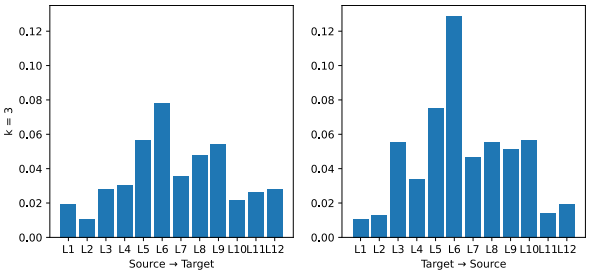


Figure 4: Distribution against layer index of the location of the top $k = 3$ most contributional heads to the distinction of pairs of relations.

probability ratio, and grouped them by layer index. For different values of k, we found that the most distinctive heads were predominantly located in layers of average depth. It appears for example that heads in Layer 6 of the TS attention often contribute the most to determining between two relations. This is also the case for layers 5 to 9 in the ST attention. For $k = 3$, for example, 13% of the top three heads are located in layer 6 of the TS attention. See Figure 4. We could also notice the imbalance in favor of TS attention for holding the top k heads for low values of k. This imbalance decreases as k increases. (See appendix B.)

5 Conclusion

Pre-trained LMs can, to some extent, code semantic relations in their attention mechanism without need of specialization. Bidirectional networks, as RoBERTa, show better ability to distinguish between different semantic roles than conditional networks, as GPT-2. Linear methods used in this work unveils an important fact. Pre-trained LMs encode not only the syntactic structure, but also the semantic structure of the text so that it can be exploited in a linear fashion.

References

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract Meaning Representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria. Association for Computational Linguistics.

Austin Blodgett and Nathan Schneider. 2021. Probabilistic, Structure-Aware Algorithms for Improved Variety, Accuracy, and Coverage of AMR Alignments. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3310–3321, Online. Association for Computational Linguistics.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does BERT look at? An analysis of BERT's attention. In *Proceedings of the 2019 ACL workshop BlackboxNLP: Analyzing and interpreting neural networks for NLP*, pages 276–286, Florence, Italy. Association for Computational Linguistics.

Marco Damonte, Shay B. Cohen, and Giorgio Satta. 2017. An Incremental Parser for Abstract Meaning Representation. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 536–546, Valencia, Spain. Association for Computational Linguistics.

Donald Davidson. 1969. The individuation of events. In Nicholas Rescher, editor, *Essays in Honor of Carl G. Hempel*, pages 216–34. Reidel.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the north American chapter of the association for computational linguistics: Human language technologies, volume 1 (long and short papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Jeffrey Flanigan, Sam Thomson, Jaime Carbonell, Chris Dyer, and Noah A. Smith. 2014. A Discriminative Graph-Based Parser for the Abstract Meaning Representation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1426–1436, Baltimore, Maryland. Association for Computational Linguistics.

Kevin Knight, Bianca Badarau, Laura Baranescu, Claire Bonial, Madalina Bardocz, Kira Griffitt, Ulf Hermjakob, Daniel Marcu, Martha Palmer, Tim O'Gorman, and Schneider Nathan. 2020. Abstract meaning representation (amr) annotation release 3.0 ldc2020t02. *Linguistic Data Consortium*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. ArXiv:1907.11692 [cs].

Ziyang Luo. 2021. Have Attention Heads in BERT Learned Constituency Grammar? In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 8–15, Online. Association for Computational Linguistics.

Joakim Nivre. 2008. Algorithms for Deterministic Incremental Dependency Parsing. *Computational Linguistics*, 34(4):513–553.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The Proposition Bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, volume 30. Curran Associates, Inc.

Chuan Wang, Nianwen Xue, and Sameer Pradhan. 2015. A Transition-based Algorithm for AMR Parsing. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 366–375, Denver, Colorado. Association for Computational Linguistics.

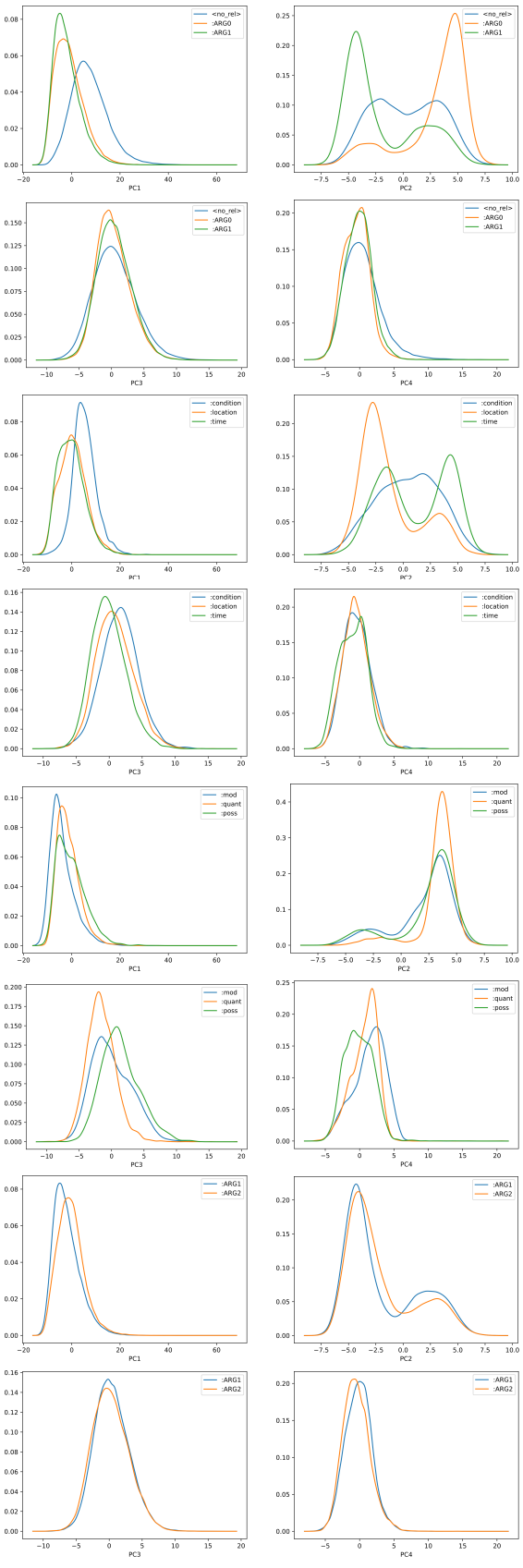
Sheng Zhang, Xutai Ma, Kevin Duh, and Benjamin Van Durme. 2019. AMR parsing as sequence-to-graph transduction. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pages 80–94, Florence, Italy. Association for Computational Linguistics.

Jiawei Zhou, Tahira Naseem, Ramón Fernandez Astudillo, Young-Suk Lee, Radu Florian, and Salim Roukos. 2021. Structure-aware fine-tuning of sequence-to-sequence transformers for transition-based AMR parsing. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pages 6279–6290, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

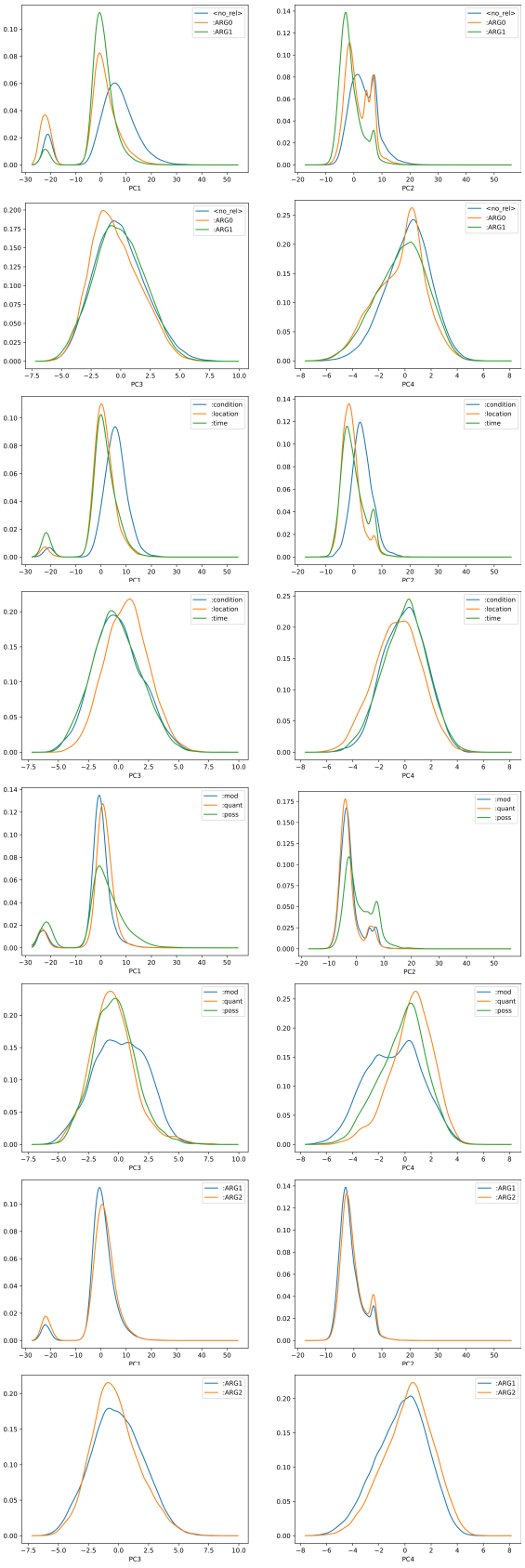
A Plots of the densities of different relations

The following figures are plots of the densities (estimated through kernel density estimation) of different relations, projected onto the first four principal components.

A.1 RoBERTa Language Model

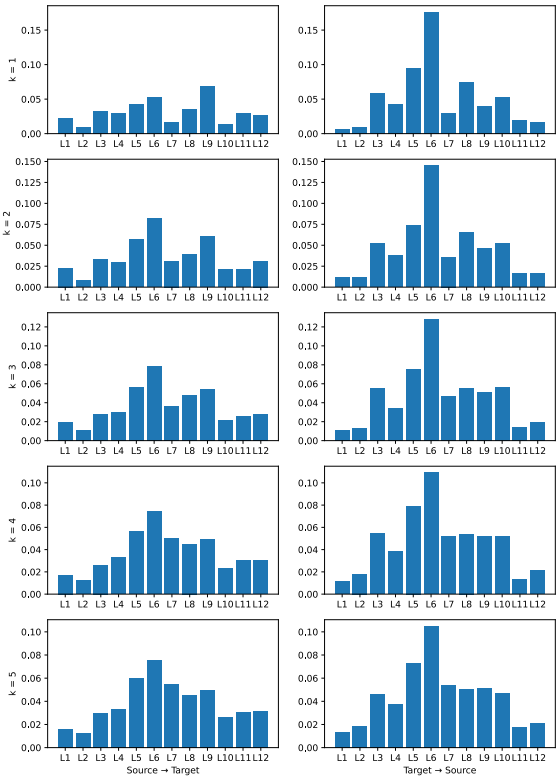


A.2 GPT-2 Language Model



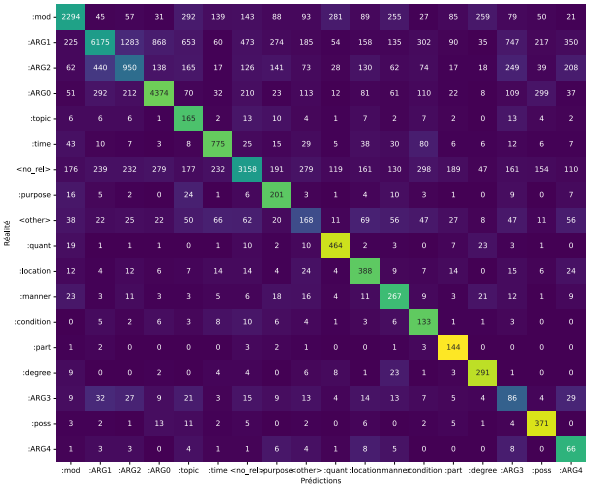
B Evolution of the distribution of the top k heads

The following figures present the evolution of the distribution against layer index of the location of the top k most contributional heads to the distinction of pairs of relations. As k increases from 1 to 5, the imbalance in favor of the TS attention decreases.

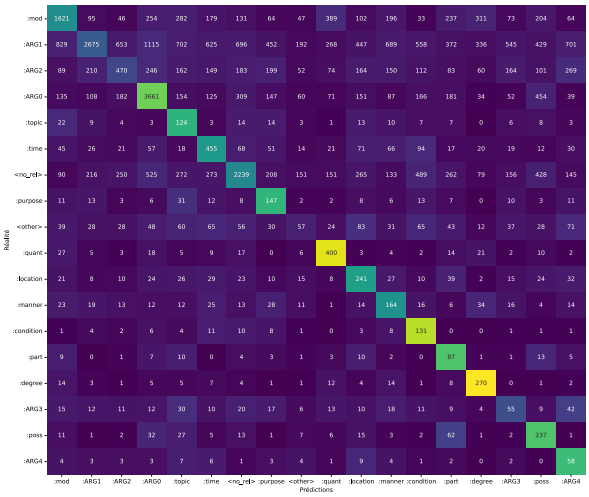


C Confusion Matrices of the Logistic Classifiers

C.1 Confusion Matrix for RoBERTa LM



C.2 Confusion Matrix for GPT-2 LM



C.3 Confusion Matrix for Augmented GPT-2 LM