*Proceedings of the 13th Conference on Language Resources and Evaluation (LREC 2022)*, pages 4040–4050
Marseille, 20-25 June 2022


# Attention Understands Semantic Relations

**Anastasia Chizhikova**[♠♡] **Sanzhar Murzakhmetov**[♠]
**Oleg Serikov**[‡♠♡] **Tatiana Shavrina**[‡$] **Mikhail Burtsev**[♠‡]
[♡] HSE University [♠] DeepPavlov lab, MIPT [‡] AIRI [$] SberDevices
Moscow, Russia
apchizhikova@edu.hse.ru, murzakhmetov.s@phystech.edu, oserikov@hse.ru
shavrina@airi.net, burtcev.ms@mipt.ru

**Abstract**

Today, natural language processing heavily relies on pre-trained large language models. Even though such models are criticised for poor interpretability, they still yield state-of-the-art solutions for a wide range of very different tasks. While many probing studies have been conducted to measure the models awareness of grammatical knowledge, semantic probing is less popular. In this work, we introduce a probing pipeline to study how semantic relations are represented in transformer language models. We show that in this task, attention scores express the information about relations similar to the layers' output activations despite their lesser ability to represent surface cues. This supports the hypothesis that attention mechanisms focus not only on syntactic relational information but semantic as well.

**Keywords:** ontology extraction, knowledge probing, semantic probing, explainable AI (XAI), language models interpretation, bertology

## 1. Introduction

Present-day monopoly of foundation language models in the majority of NLP tasks forces researchers and practitioners to rely on popular large models without genuinely understanding the models' behaviour. The development of such models, regardless of impressive results on the existing benchmarks (Wang et al., 2019; Shavrina et al., 2020), leads to an interpretability crisis too. For example, general pre-trained models are often criticised for memorising things instead of generalising over the training knowledge.

To enlighten the mechanisms driving these black-box models' behaviour, a research area of models interpretation has developed. Probing studies were conducted outlining the limits to which the learned behaviour of the existing language models agrees with those developed by research linguists (Conneau and Kiela, 2018). The majority of probing works address grammar, while semantic and factual knowledge remains largely understudied (Rogers et al., 2020).

Given the above, there is a gap in understanding of how models capture factual and semantic knowledge. By interpreting the models which link texts to the existing ontologies and databases, one could narrow this gap: models are expected to use factual and semantic knowledge to solve downstream tasks efficiently. The interpretation of such a model, namely a Relation Extraction model, is the key narrative of the present paper.

In this work, we introduce an approach to studying language models' factual and semantic knowledge representations. We do so by constructing an intentionally naive pipeline of Relation Extraction and then interpreting its steps. We choose the BERT (Devlin et al., 2018) model as an object of our study, as it is the model which is the most covered with interpretability works and a reasonable baseline for the majority of current downstream tasks. We employ a probing methodology to study if the BERT's intermediate inference steps provide enough information to identify the WikiData (Vrandečić and Krötzsch, 2014) relations over tokens on the T-REx dataset (Elsahar et al., 2018). For this end, we go beyond layer-wise probing and perform the careful interpretability analysis of self-attention mechanisms, thus extending the existing knowledge (Kovaleva et al., 2019) about the semantic abilities of these core transformer building blocks.

We show that the results of the layer-wise self-attention weights probing agree with those of the classical probing of layers activations. A closer analysis of probing classifiers reveals patterns which strongly overlap with human clustering of such relations. We thus contribute to the ongoing discussion (Bender et al., 2021) on the transformer's ability to meaningfully generalise over the pre-training data.

Our contributions, therefore, can be summarised as follows: (i) we provide the methodology for inspecting the self-attention models' linguistic knowledge. It comprises building and then analysing a simple interpretable algorithm which follows the core concepts and a base model of the reasonable baseline for a linguistically challenging task, (ii) we enrich the diagnostic probing methodology with the ability to inspect relational and primarily semantic relational knowledge, (iii) we extend the existing knowledge about self-attention semantic relations awareness by analysing novel relation classes.

We make our experiments and results available at [1]

[1] https://github.com/deepmipt/kg-extracting-probing

## 2. Related Work

### 2.1. Probing Literature

Several methodological branches can be seen in the probing community. In our research, we employ diagnostic classification techniques. At the same time, our research is strongly related to the body of attention interpretation papers — we took our inspiration in Hewitt and Manning (2019), and our results extend the findings presented in Kovaleva et al. (2019). To keep probing studies reliable, the notion of selectivity and control tasks was introduced in Hewitt and Liang (2019), which we employ in our study.

**Diagnostic Classification Probing** Diagnostic classification probing has gained its popularity for interpreting machine translation models (Alain and Bengio, 2016) and is now successfully applied to interpret the language models.

With language models, a diagnostic classification study consists of approximating the bond between objects' embeddings and linguistic property values. Studies are often conducted layer-wise, involving intermediate layers' embeddings, as opposed to downstream tasks, which mostly rely on the final layer embeddings. Such studies shed light on how well the linguistic knowledge is captured by language models and how it is distributed across them.

In their SentEval tool, Conneau and Kiela (2018) introduce ten probing tasks to study representations of linguistic properties in language models. While the SentEval tool has been originally applied to the sentence embeddings in the LSTM models, the tool has been successfully ported to study transformer models as well (Mikhailov et al., 2021). The authors of LINSPECTOR (Şahin et al., 2020) toolkit extend the diagnostic probing methodology to employ contextualised token embeddings. The probing study of semantic awareness of BERT has previously been conducted in Tenney et al. (2019). Here, they introduce the 'edge probing' approach, which uses embeddings of tokens of a span to vectorise the relational linguistic structures.

While judging the models' awareness of properties by their behaviour on data, one should keep in mind that this behaviour depends on the properties of both model and dataset. To estimate the impact of these factors, Hewitt and Liang (2019) introduce the *selectivity* term, assigning the high selectivity score to the approaches which highlight the impact of the model. To maintain selectivity, the control tasks methodology has been introduced. It can be illustrated by an example of a random initialization control task. When probing both a pre-trained and randomly initialised model, a selective approach will show significant differences in scores since only the pre-trained model has reliable knowledge. Instead, a poorly selective approach will be biased by the impact of the dataset, which stays the same in both experiments.

**Probing for Relations in Attention** Not only layers' activations embeddings are involved in probing studies but self-attention weights too. In Hewitt and Liang (2019), the authors propose a non-parametric approach to restoring syntactic trees from attention maps. To do so, for a given sentence, they treat tokens and their respective self-attention weights as a weighted graph. The deterministic procedure is then applied to build a dependency tree from such a graph. The received trees are shown to be interpretable from the linguistic point of view, supposing that self-attention mechanisms capture relational knowledge at the syntactic level.

The semantic awareness of BERT models' attention weights has been studied in Kovaleva et al. (2019). The authors provide an extensive study of self-attention mechanisms behaviour in transformer language models, conducting qualitative and quantitative studies of emerging self-attention patterns. They show that while BERT shows relatively high probing scores when identifying frame-semantic relations, there is no evidence of attention maps specialization on particular types of relations. In their work, the authors highlight the necessity to investigate a broader range of relations, which we do in the present work.

In Wallat et al. (2021), the authors conclude that intermediate layers of language models may contain knowledge which is forgotten by the final layers. Thus, the knowledge is distributed unevenly among the layers. In order to learn to extract it from attention maps, we need to study this distribution better. We address this curiosity by conducting a close study of individual attention heads' role in solving probing tasks.

### 2.2. Relation Extraction Literature

Various recent neural networks use fine-tuned models to solve the task of relation extraction from a raw sequence. For example, Liu et al. (2020) train an attention-based joint model and use a supervised multi-head self-attention mechanism to solve that task. In Xue et al. (2019), the authors introduce a focused attention model and jointly train it on entity and relation extraction tasks based on a shared task representation encoder which is transformed from BERT through a dynamic range attention mechanism, out-performing all previous models on the task. Although such supervised approaches appear to be very effective and solve the task with high quality, they often lack interpretability.

In Wang et al. (2020), the authors propose a fully unsupervised method of relation extraction — the MaMa (Match and Map) approach. First, they obtain a set of candidates through a beam search algorithm over an attention matrix from the last layer (averaged over heads). Then they filter the triplets using several constraints (e.g., empirically chosen threshold) and map them to WikiData entities. The simplicity of the proposed efficient method leaves room for certain stages of improvement. See section 5 for relevant insights from our experiments.

In Cabot and Navigli (2021), the authors address the Relation Extraction as a sequence-to-sequence task, training a transformer (namely, BART-large) model to translate input texts into a textual representation of triplets. For example, the text *"This Must Be the Place" is a song by a new wave band Talking Heads*, containing the triplets ⟨*This Must Be the Place*, *performer*, *Talking Heads*⟩ and ⟨*Talking Heads*, *genre*, *new wave*⟩, will receive the following translation:

```
<triplet> This Must Be the Place
<subj> Talking Heads <obj> performer
<triplet> Talking Heads <subj> new
wave <obj> genre.
```

### 2.3. Data Resources

In our work, we use the WikiData (Vrandečić and Krötzsch, 2014), T-REx (Elsahar et al., 2018), and DocRED (Yao et al., 2019) resources to evaluate and interpret models.

**WikiData** The WikiData resource (Vrandečić and Krötzsch, 2014) is a knowledge base with more than 90 million data items that are connected by properties (semantic relations). This open graph contains factological knowledge from Wikipedia and other related resources.

**T-REx** The T-REx dataset was introduced in (Elsahar et al., 2018). It consists of 11 million WikiData triplets of tokens automatically aligned with about 3 million Wikipedia abstracts. For our experiments, we sampled a sub-dataset of T-REx balanced by classes, leaving only 95 sufficiently presented relations.

**DocRED** DocRED (Document-Level Relation Extraction Dataset) (Yao et al., 2019) is a relation extraction dataset built from Wikipedia and WikiData. Since the raw data for the dataset was provided by human contributors, the dataset supplies extensive distantly supervised data. We involve 2000 texts with 9852 triplets sub-sample of the DocRED corpus for our experiments.

## 3. Methods

To study the semantic awareness of BERT, we employ a two-stage workflow. We first build an easily interpretable Relation Extraction model on top of BERT. Then through the proxy of this model, we analyse the BERT's semantic knowledge.

### 3.1. Interpretable Relation Extraction model

Following (Wang et al., 2020), we formulate the Relation Extraction task as identifying triplets of tokens in text and labelling them with the respective relations Ids. Each triplet represents one entry of particular semantic relation (in our case, the relations are WikiData properties) in a given text. Each triplet consists of *head*, *tail*, and *rel* (in short, *h*, *t*, *r*). For example, the sentence *Moscow is the capital of Russia* would have one annotated triplet ⟨*Russia*, *Moscow*, *capital*⟩ with label `P36` (*capital*). We only involve the triplets that have an explicitly expressed form of a *rel* token in the text. This ignores the case when the relation between two entities might not be expressed in any token of the text at all. Such opaque relation triplets are always filtered out (e.g. from the T-REx dataset) in our study.

The algorithm works on a sentence level. To identify the triplets, we employ a binary classifier which predicts whether a triplet encodes a relation in the sentence or not. We select the triplets for further processing with this classifier, heuristically filtering out the linguistically implausible triplets. These triplets are then fed into a multiclass classifier which is trained to predict the relation Id for the given attentions between tokens of the triplet. The confidence of both classifiers was used to filter out the unlikely relation triplets.

**Triplet Candidates Filtering Heuristics** Since all of the target triplets follow almost the same syntactic patterns and are usually expressed by a limited group of parts of speech, we use POS filtering to select the triplets with plausible POS tags. Only noun groups and pronouns are allowed to be *head* and *tail* candidates, and only verbs and noun groups could be generated as a potential *rel* token. We use the NLTK (Loper and Bird, 2002) chunker to identify noun groups. [2]

**Triplets Vectorization** Each triplet is represented by a vector which we obtain by collecting attention weights between the tokens of the triplet. The weights from all layers and all heads of the BERT model are taken and then concatenated into one vector. Since there exist six types of pairwise weights for every triplet of the tokens in every attention matrix (*head-rel*, *rel-tail*, etc.), we end up having a vector of length 864 (6 types of attention $\times$ 12 layers $\times$ 12 heads). Thus, every attention weight is treated as a feature of the triplet. If a *head*, *tail* or *rel* token consists of multiple subtokens after the BPE-tokenization step, their attention weights were aggregated with mean pooling.

**Binary Classifier** uses logistic regression to identify if there is a semantic relation between the tokens. Since we generate many candidates for every input sequence, we pay attention to the triplets which the model is most confident of.

The model is trained on a balanced subsample inferred from the T-REx. The original T-REx annotations serve as a positive class, and the triplets for the negative class are randomly sampled from the same sentences. This sampling strategy assumes that all true triplets are annotated in the T-REx dataset. Carefully inspecting the dataset, we find sentences lacking several triplets annotation; thus, our classifier is trained on a slightly shifted dataset.

**Multiclass Classifier** logistic regression is used to further predict the relation Id for the chosen triplets.

[2] We have experimented with Spacy as well, with worse overall performance

**The Complete Pipeline.** The purpose of training two separate classifiers is twofold: first, it simply increases the overall predicting ability of the full pipeline; second, distinguishing knowledge detection from its further classification allows for more interpretation strategies.

The whole pipeline is summarized in Figure 1.

During our experiments, we score the pipeline twice. First, we evaluate the average F1-score for both classifiers on the respective previously held-out parts of the training data. This evaluation allows estimating individual stages' performance in perfect conditions. Second, we evaluate our whole Relation Extraction pipeline on the golden standard data.

### 3.2. Using RE Model to Analyse BERT

The proposed Relation Extraction model can be used to analyse the BERT's semantic knowledge. The performance of the binary classifier proxies the simplicity of relation tokens identification. The study of generalizations made by the multiclass classifier complements the analysis of the whole pipeline. It approximates how informative attention weights are from a more detailed perspective. We inspect the multiclass model following the diagnostic probing methodology and support our experiments with the random initialization control task.

The classifiers' features importances provide us with the mapping of relations onto the multidimensional space. The attention heads induce the axes, and the coordinates for every relation are these heads' importance weights. To better understand the behavioural patterns of the attention heads, we inspect the relations groupings in this space and estimate the individual axes' importances for the RE task.

## 4. Results

### 4.1. Relation Extraction Performance

**Classifiers** Table 3 summarises the classifiers' performance on a test set (macro averaged by relations for the multiclass classifier).

**Probes Selectivity** The randomly initialised BERT shows poor score in our experiment (Table 3). The probing models are unable to learn and generalise connections between the tokens, suggesting that there is neither hidden knowledge nor significant leakages of dataset surface structure through the model layers. This allows us to call our probes selective.

**Triplets Extraction Scores and Threshold Parameter** Table 1 presents the results of our triplets extraction algorithm on a raw text with different confidence thresholds applied. This experiment reveals that both binary and multiclass models' confidences are useful to predict triplets.

**Extraction Quality Analysis** We find a significant mismatch between the measured scores of the classification models and the entire pipeline evaluation. While models operate with the triplets with a reasonably high score, the metrics of the relation extraction on raw texts are significantly lower.

| Binary Threshold | Multi Threshold | Pr* | Rec* | F1 |
|---|---|---|---|---|
| **0.7** | **0.2** | 0.135 | 0.279 | **0.176** |
| **0.8** | **0.2** | 0.145 | 0.241 | 0.174 |
| **0.9** | **0.2** | 0.169 | 0.178 | 0.163 |
| **0.7** | **0** | 0.096 | 0.366 | 0.149 |
| **0.8** | **0** | 0.108 | 0.291 | 0.155 |
| **0.9** | **0** | 0.136 | 0.202 | 0.155 |

Table 1: Results of the triplets prediction on a raw text with different threshold parameters. The best extraction scores are in bold.

As the number of tokens in a sentence increases, the number of generated candidates grows exponentially (see Table 2). In contrast, the number of the annotated target triplets remains about the same. As a result, even the classifier that predicts meaningful triplets with $0.86$ accuracy on a balanced dataset ends up over-generating triplets on the real data, worsened even more by the imbalance of classes on the real data.

| Size Bucket | Mean Candidates | Pr | Rec | F1 |
|---|---|---|---|---|
| (0, 10) | 20 | 0.4328 | 0.4873 | 0.4584 |
| (10, 20) | 118 | 0.1628 | 0.3043 | 0.2121 |
| (20, 40) | 841 | 0.0611 | 0.2184 | 0.0954 |
| (40, 70) | 4169 | 0.0190 | 0.1570 | 0.0338 |
| (70, 100) | 17601 | 0.0057 | 0.1005 | 0.0108 |
| (100, 150) | 61166 | 0.0002 | 0.0104 | 0.0005 |

Table 2: Bucket statistics.

| Attention | Classifier | Pr | Rec | F1 | Acc* |
|---|---|---|---|---|---|
| BERT | Binary | | | | 0.902 |
| Random BERT | Binary | | | | 0.498 |
| BERT | Multi | 0.867 | 0.861 | 0.863 | |
| Random BERT | Multi | 0.026 | 0.024 | 0.018 | |

Table 3: Classifiers test scores.

### 4.2. Comparison with the existing Relation Extraction models

To calibrate the degree of belief in our model, we compare its performance with another Relation Extraction model, REBEL (Cabot and Navigli, 2021) (Table 4). Both models are scored on two datasets, T-REx and DocREd. On T-REx, our method outperforms REBEL, primarily due to the ability to process pronouns. On the DocREd, the performance of our method drops significantly due to the over-generation described in 4.1.

*Pr stands for Precision, Rec stands for Recall, Acc stands for Accuracy

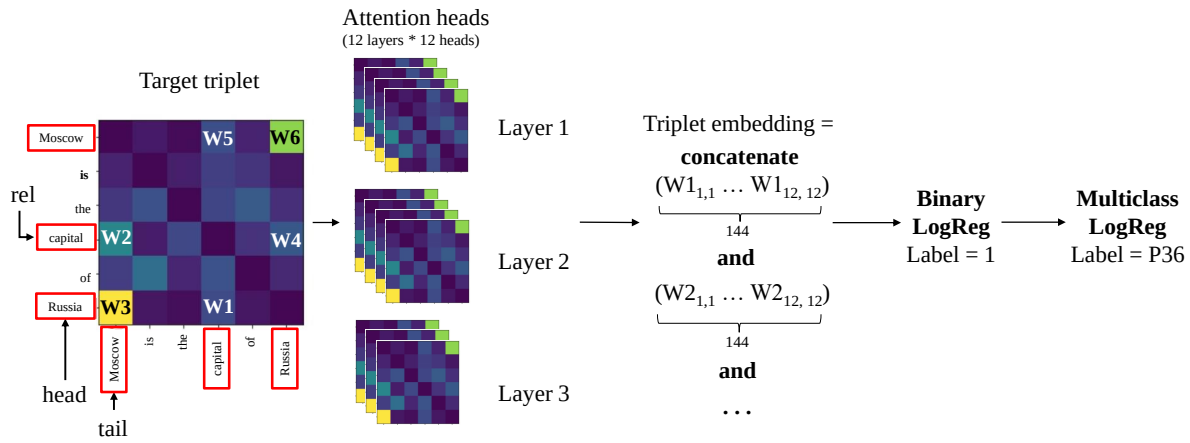


Figure 1: Summary of proposed relation classification pipeline. A potential triplet is encoded by concatenation of attention scores between all components from the BERT model. Resulting feature vector is sequentially classified by binary and multiclass logistic regression classifiers.

See Appendix A for the example of knowledge graphs provided by both approaches for the exact text. On both datasets we heuristically rank the output of binary model according to the confidence.

| Model | Dataset | Pr | Rec | F1 |
|---|---|---|---|---|
| **Our** | T-REx | 0.220 | 0.534 | **0.312** |
| REBEL | T-REx | 0.223 | 0.375 | 0.276 |
| Our | DocREd | 0.259 | 0.175 | 0.208 |
| **REBEL** | DocREd | 0.594 | 0.437 | **0.503** |

Table 4: Performances comparison with DocREd.

### 4.3. Self-Attention Types Importance

The existence of six attention weight types for each triplet yields multiple available vectorization strategies, simply allowing us to exclude weights of a particular type from pooling. We conduct 63 experiments with all possible combinations of attention types to inspect which types of attention provide less information than the others. Our experiments show that the setup that uses all types of attention weights outperforms any other weights combination. The increase in the attention types involved leads to the growth of the prediction score. Table 5 shows top-5 attention types combinations without ranking candidates by confidence of binary model.

Figure 2 depicts the importance of attention types for the whole pipeline. Although the importance of all types is slightly different, none of these can be neglected, as there is no dramatic difference between them.

Figure 3 depicts the extraction F1-scores for every semantic relations for two attention feature combination strategies. One includes weights between only *head* and *tail* token and another between only *tail* and *rel* tokens, with the red curve showing the performance of the full six weight types-based model. This allows us to observe how the role of different attention features in the predictive ability of the pipeline. The model which only uses the *tail-rel* attention feature outperforms the *head-tail* feature model. The experiments with other attention types support this. The models which rely on the relation tokens attention weights (e.g. *t-r*, *t-h*) perform better than those that do not. This fact could be explained by the fact that attention weights which feature relation tokens are the most significant in relation prediction tasks as they carry more information about the semantic relation between two entities. For example, in the case when *head* and *tail* tokens are names of people, triplets with relations like `P22` (*mother*), `P25` (*father*), `P26` (*spouse*), etc., cannot be disambiguated without the aid of a relation token.

Similar plots for all 63 experiments are provided in the repository: `https://github.com/deepmipt/kg-extracting-probing/blob/main/data/meta/attentions_all_cropped.pdf`

| Attention types | Pr | Rec | F1 |
|---|---|---|---|
| h-r r-t h-t r-h t-r t-h | 0.135 | 0.279 | 0.176 |
| h-r r-t h-t r-h t-r | 0.133 | 0.274 | 0.173 |
| h-r r-t r-h t-r t-h | 0.132 | 0.267 | 0.170 |
| h-r r-t h-t t-r t-h | 0.132 | 0.26 | 0.169 |
| h-r h-t r-h t-r t-h | 0.126 | 0.261 | 0.164 |

Table 5: Top-5 best combinations of attention type by triplets prediction scores.

### 4.4. Layers Importance

To see how the knowledge is distributed across the layers and how each of them affects the results of the Relation Extraction, we conducted 12 experiments, zeroing

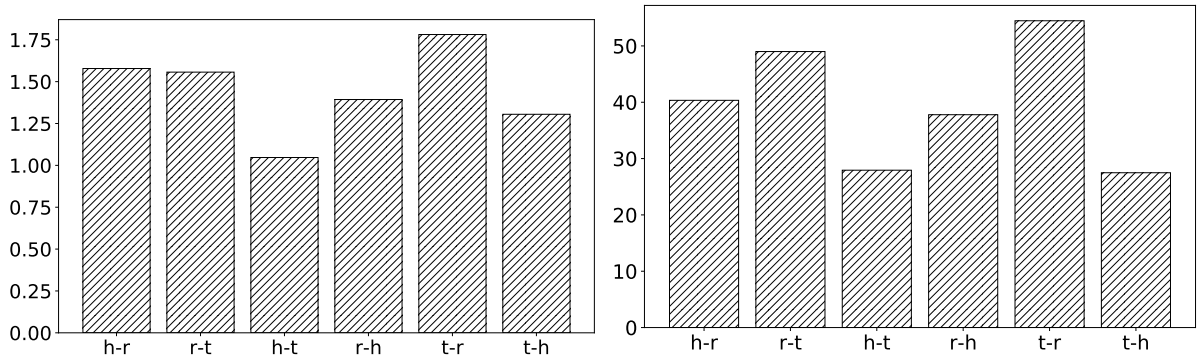


Figure 2: Feature importance values averaged by the attention type for the binary (left) and multiclass (right) classifier.

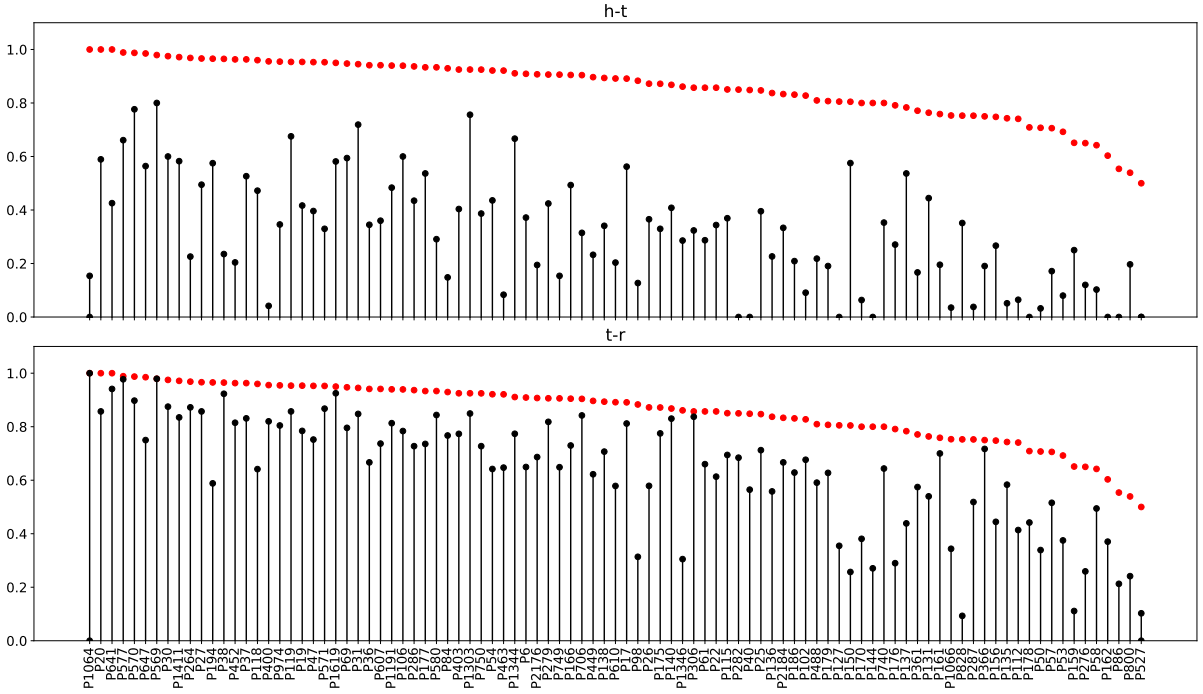


Figure 3: F1 stem plots for head to tail (h-t, top) and tail to relation (t-r, bottom) attention features for different relations (x-axis), red line represents the baseline model with six features.

the attention weights for all layers except a particular layer every time.

Figure 4 illustrates how the extraction and relation type prediction scores change layer-wise for different types of semantic relations. The layers are shown to differ in terms of the ability to predict the relation type. Unlike in (Tenney et al., 2019), we discover that not only higher layers might be responsible for encoding the semantic features of the sequence. For instance, both of these plots show layers 4 and 10 to be notably semantically informative.

We also assess the classical diagnostic probing study to compare the attention weights scores with those of BERT layers units activations. Again, we perform the experiment layer-wise. The BERT layers embeddings achieve the performance of $> 99\%$. Despite that, the BERT layers probing (see Figure 6) reveals the same pattern as the attention heads probing — the first layers of the model are more semantically informative than the last ones.

### 4.5. Clusterisation of Relations

To study the semantic relations hierarchy made by BERT, we inspect the multiclass model which features all six types of attention weights. The analysis of the feature importance weights of the multiclass classifier shows that semantically close types of relations have similar weights in the logistic regression model (that is, they are similarly encoded in the attention mechanism). Furthermore, the hierarchy of these importance vectors yields a data-driven semantic classification of WikiData properties we used.

Figure 7 presents the results of agglomerative cluster-

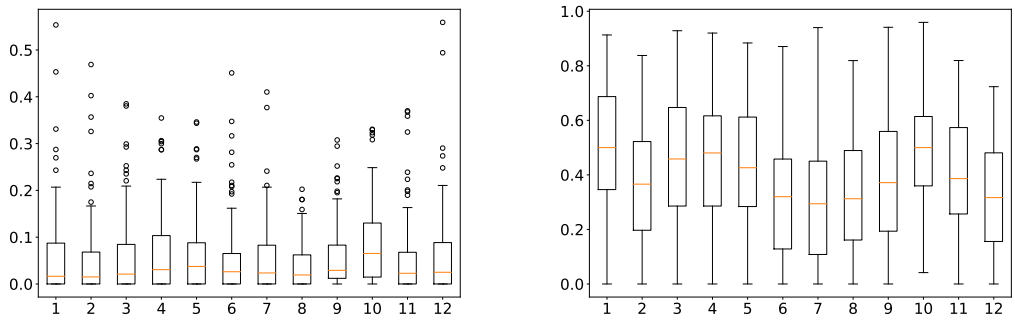


Figure 4: Attention layers probe. F1-scores for triplets extraction task on a raw dataset (left) and F1-scores of relation label prediction for the extracted triplets (right).

ing with cosine distance. We observe that the relation types are logically organised into semantic groups, which means that the complicated way of how knowledge is structured inside language models is not purely stochastic but can be interpreted. We find no strict mapping between the attention heads and the semantic relation types.

Multilabel classifiers tend to make two types of mistakes. First, the performance on more abstract types of relations (e.g. *has part*) is lower than on purely factological ones (e.g. *place of burial*). Second, semantically close types of relations such as *writer*, *composer* and *screenwriter* are often confused. Figure 5 shows how the quality of relation label prediction increases when joining the closest relation classes into one based on relation vectors similarity yielded by agglomerative clustering. As more classes get merged, the score expectedly grows, though we never achieve a perfect score. Notably, following the classes hierarchy provided by WikiData itself in the same experiment leads to F1-score drop.

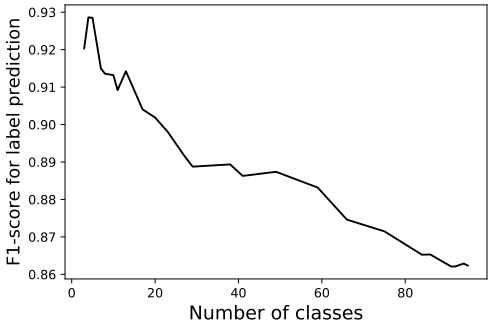


Figure 5: Relation prediction F1-scores for different number of classes after clustering.

## 5. Conclusion & Future Work

In this study, we introduce a novel approach to interpretation of language models. In this approach only attention scores are taken into account to encode BERT's

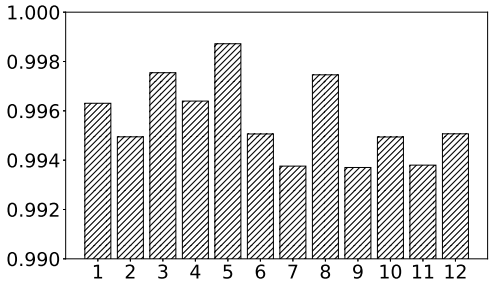


Figure 6: Layer-wise probing. F1-scores for relation label prediction on the BERT layers embeddings.

awareness of semantic relations. We build a two-stage Relations Extraction downstream model, which shows that attributing a relation to a particular triplet given attention features can be performed quite easily. To improve our extraction method in the future, we need to analyse the syntactic nature of the relation triplets thoroughly.

We find that semantic relations of different types are encoded with a combination of attention weights provided by different heads. We show that attention weights are not as informative as layers' units activations but provide a reliable, straightforward approach to ranking the layers' awareness of relational linguistic features. The proposed interpretation methodology is explicit and straightforward, as opposed to the edge probing strategy (Hewitt and Manning, 2019).

We diagnose the impact of different attention weights and layers on the model's ability to predict the exact type of a semantic relation. We conclude that none of the layers and attentions must be neglected while developing an unsupervised approach to relation extraction. This could be used to improve the unsupervised Relations Extraction technique presented in Wang et al. (2020), whose authors employ only the final layer attention weights.

While inspecting the attention heads' specialization, we discover that there are no individual relation-

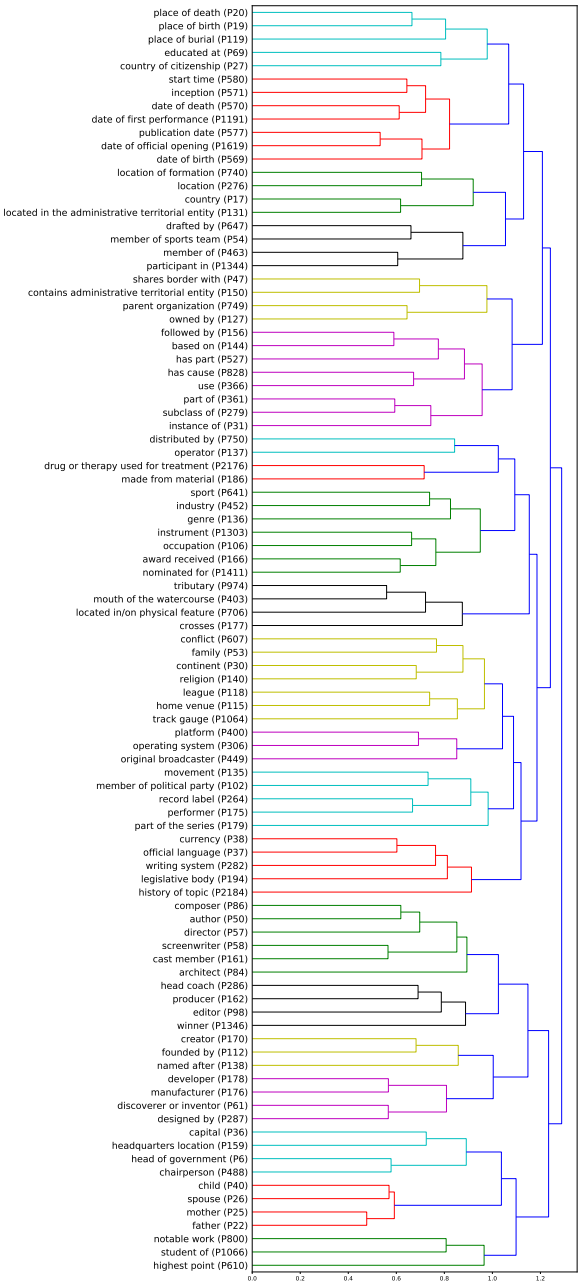


Figure 7: Agglomerative clustering shows interpretable hierarchical grouping of semantic relations.

specific heads, yet one could meaningfully group relations by the heads' relevance for them. While the heads' specialization patterns remain opaque, their overall behaviour correlates with linguists' judgements about the semantics. We assume two main factors to impact the specialization of attention heads. The first of them is *head* and *tail* entities types (e.g. the upper-blue branch in Figure 7 is composed of semantic relations which describe a person in time and space), the second is the syntactic relation between *head*, *tail* and *rel* tokens. We find no correlation between the model's ability to extract triplets and the syntactic distance between the involved entities. Nevertheless, the role of syntax (e.g. the type of construction which the relation is usually expressed by) on the relation's linkage requires further investigation.

We were unable to reach the perfect score by continuously merging the relations classes according to the hierarchy proposed by WikiData. Although intuitive, it might not coherently correspond to the way how language models see semantic relations. Thus, our interpretation mechanism might shed light on this difference and propose a way to figure out a new relation classification to use in Knowledge Graphs. Similarly, during the algorithms errors analysis, we discover that the existing datasets are inaccurate in terms of precision and recall, resulting in the evaluation scores flaw.

## 6. Acknowledgements

This work was supported by a grant for research centers in the field of artificial intelligence, provided by the Analytical Center for the Government of the Russian Federation in accordance with the subsidy agreement (agreement identifier 000000D730321P5Q0002) and the agreement with the Moscow Institute of Physics and Technology dated November 1, 2021 No. 70-2021-00138

## 7. Bibliographical References


Alain, G. and Bengio, Y. (2016). Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*.

Bender, E. M., Gebru, T., McMillan-Major, A., and Shmitchell, S. (2021). On the dangers of stochastic parrots: Can language models be too big?. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 610–623.

Cabot, P.-L. H. and Navigli, R. (2021). Rebel: Relation extraction by end-to-end language generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2370–2381.

Conneau, A. and Kiela, D. (2018). Senteval: An evaluation toolkit for universal sentence representations. *arXiv preprint arXiv:1803.05449*.

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*.

Hewitt, J. and Liang, P. (2019). Designing and interpreting probes with control tasks. *arXiv preprint arXiv:1909.03368*.

Hewitt, J. and Manning, C. D. (2019). A structural probe for finding syntax in word representations. 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 4129–4138.

Kovaleva, O., Romanov, A., Rogers, A., and Rumshisky, A. (2019). Revealing the dark secrets of BERT. *CoRR*, abs/1908.08593.

Liu, J., Chen, S., Wang, B., Zhang, J., Li, N., and Xu, T. (2020). Attention as relation: Learning supervised multi-head self-attention for relation extraction. pages 3759–3765, 07.

Loper, E. and Bird, S. (2002). Nltk: The natural language toolkit. *arXiv preprint cs/0205028*.

Mikhailov, V., Taktasheva, E., Sigdel, E., and Artemova, E. (2021). Rusenteval: Linguistic source, encoder force! *arXiv preprint arXiv:2103.00573*.

Rogers, A., Kovaleva, O., and Rumshisky, A. (2020). A primer in bertology: What we know about how bert works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Şahin, G. G., Vania, C., Kuznetsov, I., and Gurevych, I. (2020). Linspector: Multilingual probing tasks for word representations. *Computational Linguistics*, 46(2):335–385.

Shavrina, T., Fenogenova, A., Emelyanov, A., Shevelev, D., Artemova, E., Malykh, V., Mikhailov, V., Tikhonova, M., Chertok, A., and Evlampiev, A. (2020). Russiansuperglue: A russian language understanding evaluation benchmark. *arXiv preprint arXiv:2010.15925*.

Tenney, I., Das, D., and Pavlick, E. (2019). BERT rediscovers the classical NLP pipeline. *CoRR*, abs/1905.05950.

Wallat, J., Singh, J., and Anand, A. (2021). Bertnesia: Investigating the capture and forgetting of knowledge in bert. *arXiv preprint arXiv:2106.02902*.

Wang, A., Pruksachatkun, Y., Nangia, N., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. R. (2019). Superglue: A stickier benchmark for general-purpose language understanding systems. *arXiv preprint arXiv:1905.00537*.

Wang, C., Liu, X., and Song, D. (2020). Language models are open knowledge graphs. *CoRR*, abs/2010.11967.

Xue, K., Zhou, Y., Ma, Z., Ruan, T., Zhang, H., and He, P. (2019). Fine-tuning bert for joint entity and relation extraction in chinese medical text. pages 892–897, 11.


## 8. Language Resource References


Elsahar, H., Vougiouklis, P., Remaci, A., Gravier, C., Hare, J., Laforest, F., and Simperl, E. (2018). T-rex: A large scale alignment of natural language with knowledge base triples. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Vrandečić, Denny and Krötzsch, Markus. (2014). *Wikidata: a free collaborative knowledgebase*. ACM New York, NY, USA.

Yao, Y., Ye, D., Li, P., Han, X., Lin, Y., Liu, Z., Liu, Z., Huang, L., Zhou, J., and Sun, M. (2019). Docred: A large-scale document-level relation extraction dataset. *arXiv preprint arXiv:1906.06127*.

# A. Appendix

Figure 8 provides an example of our RE model inference compared with the golden baseline and the inference of the REBel algorithm for the following text.

Trump and his businesses have been involved in more than 4,000 state and federal legal actions, including six bankruptcies. Trump's political positions have been described as populist, protectionist, isolationist, and nationalist. He entered the 2016 presidential race as a Republican and was elected in an upset victory over Democratic nominee Hillary Clinton while losing the popular vote,[a] becoming the first U.S. president with no prior military or government service. The 2017{2019 special counsel investigation led by Robert Mueller established that Russia interfered in the 2016 election to benefit the Trump campaign, but not that members of the Trump campaign conspired or coordinated with Russian election interference activities. Trump's election and policies sparked numerous protests. Trump made many false and misleading statements during his campaigns and presidency, to a degree unprecedented in American politics, and promoted conspiracy theories. Many of his comments and actions have been characterized as racially charged or racist, and many as misogynistic.

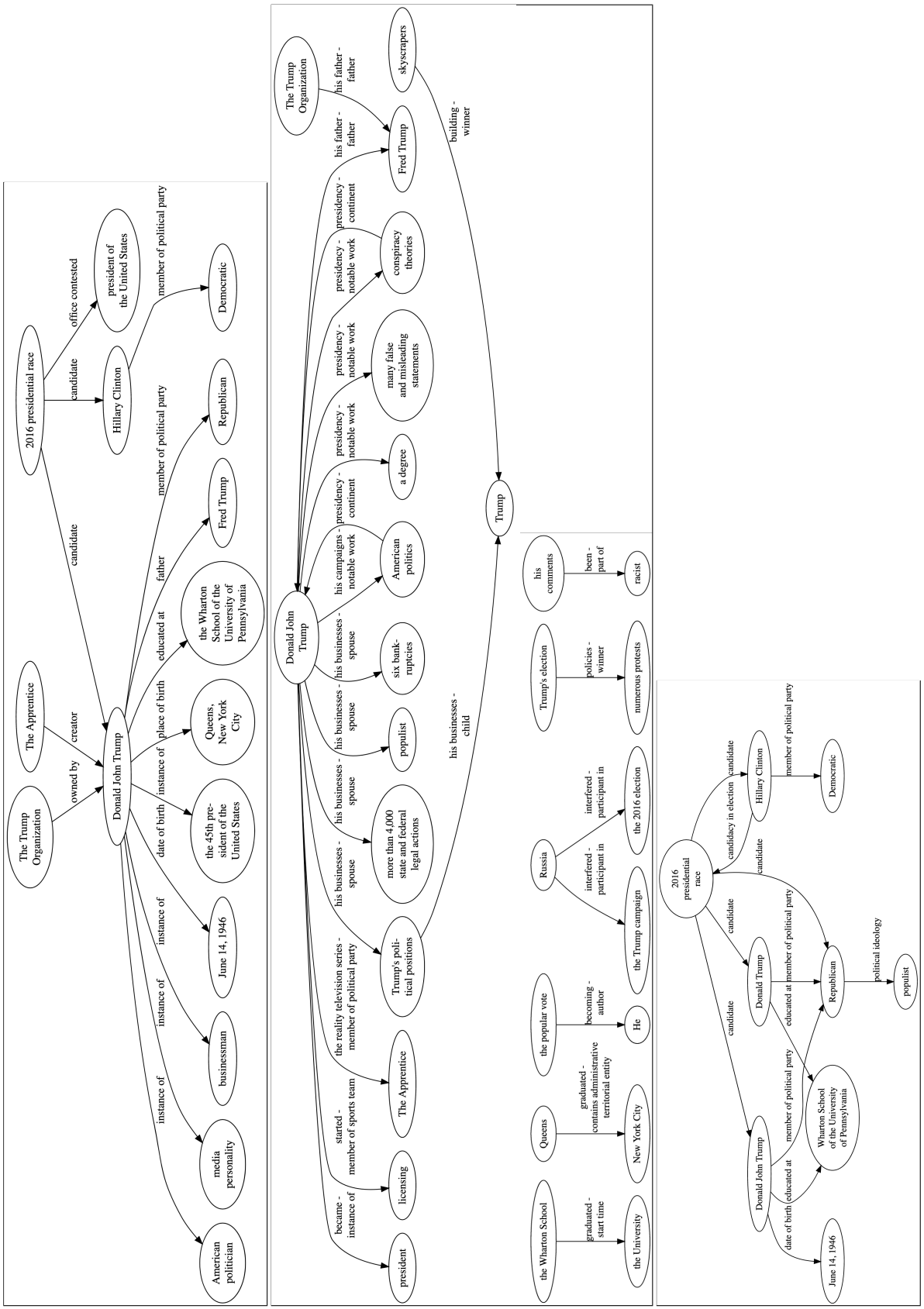


Figure 8: Knowledge graph for the Donald Trump media entry. Golden Standard (left), our algorithm (middle) and REBEL (right) versions