Acceptability Judgements via Examining the Topology of Attention Maps


1Skolkovo Institute of Science and Technology, 2AII, 3SberDevices, 4HSE University, 5Huawei Noah’s Ark lab, 6CNRS, IMJ

7Center for Information and Language Processing (CIS), LMU Munich, Germany

Abstract

The role of the attention mechanism in encoding linguistic knowledge has received special interest in NLP. However, the attention heads’ ability to judge the grammatical acceptability of a sentence has been underexplored. This paper approaches the paradigm of acceptability judgments with topological data analysis (TDA), showing that the topological properties of the attention graph can be efficiently exploited for two standard practices in linguistics: binary judgments and linguistic minimal pairs. Topological features enhance the BERT-based acceptability classifier scores by up to 0.24 Matthew’s correlation coefficient score on CoLA in three languages (English, Italian, and Swedish). By revealing the topological discrepancy between attention graphs of minimal pairs, we achieve the human-level performance on the BLIMP benchmark, outperforming nine statistical and Transformer LM baselines. At the same time, TDA provides the foundation for analyzing the linguistic functions of attention heads and interpreting the correspondence between the graph features and grammatical phenomena. We publicly release the code and other materials used in the experiments.

1 Introduction

Linguistic competence of neural language models (LMs) has emerged as one of the core sub-fields in NLP. The research paradigms explore whether Transformer LMs (Vaswani et al., 2017) induce linguistic generalizations from raw pre-training corpora (Warstadt et al., 2020b; Zhang et al., 2021), what properties are learned during task-specific fine-tuning (Miaschi et al., 2020; Merchant et al., 2020), and how the experimental results are connected to grammar and language acquisition theories (Pater, 2019; Manning et al., 2020).

One of these paradigms is centered around acceptability judgments, which have formed an empirical foundation in generative linguistics over the last six decades (Chomsky, 1965; Schütze, 1996; Scholz et al., 2021). Acceptability of linguistic stimuli is traditionally investigated in the form of a forced choice between binary categories or minimal pairs (Sprouse, 2018), which are widely adopted for acceptability classification (Linzen et al., 2016; Warstadt et al., 2019) and probabilistic LM scoring (Lau et al., 2017).

A scope of approaches has been proposed to interpret the roles of hundreds of attention heads in encoding linguistic properties (Htut et al., 2019; Wu et al., 2020) and identify how the most influential ones benefit the downstream performance (Voita et al., 2019; Jo and Myaeng, 2020). Prior work has demonstrated that heads induce grammar formalisms and structural knowledge (Zhou and Zhao, 2019; Lin et al., 2019; Luo, 2021), and linguistic features motivate attention patterns (Kovaleva et al., 2019; Clark et al., 2019). Recent studies also show that certain heads can have multiple functional roles (Pande et al., 2021) and even perform syntactic functions for typologically distant languages (Ravishankar et al., 2021).

Our paper presents one of the first attempts to analyze attention heads in the context of linguistic acceptability (LA) using topological data analysis (TDA; Chazal and Michel, 2017). TDA allows for exploiting complex structures underlying textual data and investigating graph representations of Transformer’s attention maps. We show that topological features are sensitive to well-established LA contrasts, and the grammatical phenomena can be encoded with the topological properties of the attention map.

*Equal contribution.

1github.com/danchern97/tda4la

We also refer the reader to works on computational topology and its applications (Barannikov, 1994; Zomorodian, 2001; Edelsbrunner and Harer, 2010; Carlsson and Vejdemo-Johansson, 2021).
The main contributions are the following: (i) We adapt TDA methods to two standard approaches to LA judgments: acceptability classification and scoring minimal pairs (§3). (ii) We conduct acceptability classification experiments in three Indo-European languages (English, Italian, and Swedish) and outperform the established baselines (§4). (iii) We introduce two scoring functions, which reach the human-level performance in discriminating between minimal pairs in English and surpass nine statistical and Transformer LM baselines (§5). (iv) The linguistic analysis of the feature space proves that TDA can serve as a complementary approach to interpreting the attention mechanism and identifying heads with linguistic functions (§4.3, §5.3, §6).

2 Related Work

2.1 Linguistic Acceptability

Acceptability Classification. Early works approach acceptability classification with classic ML methods, hand-crafted feature templates, and probabilistic syntax parsers (Cherry and Quirk, 2008; Wagner et al., 2009; Post, 2011). Another line employs statistical LMs (Heilman et al., 2014), including threshold-based classification with LM scoring functions (Clark et al., 2013). The ability of RNN-based models (Elman, 1990; Hochreiter and Schmidhuber, 1997) to capture long-distance regularities has stimulated investigation of their grammatical sensitivity (Linzen et al., 2016). With the release of the Corpus of Linguistic Acceptability (CoLA; Warstadt et al., 2019) and advances in language modeling, the focus has shifted towards Transformer LMs (Yin et al., 2020), establishing LA as a proxy for natural language understanding (NLU) abilities (Wang et al., 2018) and linguistic competence of LMs (Warstadt and Bowman, 2019).

Linguistic Minimal Pairs. A forced choice between minimal pairs is a complementary approach to LA, which evaluates preferences between pairs of sentences that contrast an isolated grammatical phenomenon (Schütze, 1996). The idea of discriminating between minimal contrastive pairs has been widely applied to scoring generated hypotheses in downstream tasks (Pauls and Klein, 2012; Salazar et al., 2020), measuring social biases (Nangia et al., 2020), analyzing machine translation models (Burlot and Yvon, 2017; Sennrich, 2017), and linguistic profiling of LMs in multiple languages (Marvin and Linzen, 2018; Mueller et al., 2020).

2.2 Topological Data Analysis in NLP

TDA has found several applications in NLP. One of them is word sense induction by clustering word graphs and detecting their connected components. The graphs can be built from word dictionaries (Levary et al., 2012), association networks (Dubuisson et al., 2013), and word vector representations (Jakubowski et al., 2020). Another direction involves building classifiers upon geometric structural properties for movie genre detection (Dosli and Zadrozny, 2018), textual entailment (Savle et al., 2019), and document classification (Das et al., 2021; Werenski et al., 2022). Recent works have mainly focused on the topology of LMs’ internal representations. Kushnareva et al. (2021) represent attention maps with TDA features to approach artificial text detection. Colombo et al. (2021) introduce BARYSCORE, an automatic evaluation metric for text generation that relies on Wasserstein distance and barycenters. To the best of our knowledge, TDA methods have not yet been applied to LA.

3 Methodology

3.1 Attention Graph

We treat Transformer’s attention matrix $A^{attn}$ as a weighted graph $G$, where the vertices represent tokens, and the edges connect pairs of tokens with mutual attention weights. This representation can be used to build a family of attention graphs called filtration, i.e., an ordered set of graphs $G^{\tau_i}$ filtered by increasing attention weight thresholds $\tau_i$. Filtering edges lower than the given threshold affects the graph structure and its core features, e.g., the number of edges, connected components, or cycles. TDA techniques allow tracking these changes, identifying the moments of when the features appear (i.e., their “birth”) or disappear (i.e., their “death”), and associating a lifetime to them. The latter is encoded as a set of intervals called a “barcode”, where each interval (“bar”) lasts from the feature’s “birth” to its “death”. The barcode characterizes the persistent features of attention graphs and describes their stability.

Example. Let us illustrate the process of computing the attention graph filtration and barcodes given an Example (1).
There is snowing today.

First, we compute attention maps for each Transformer head as shown in Figure 1a-1b (left). These two heads follow different attention patterns (Clark et al., 2019): attention to the next token (Figure 1a) and to the [SEP] token (Figure 1b). Next, we represent the map as a weighted graph, and conduct the filtration procedure for a fixed set of attention weight thresholds. The edges lower than each given threshold are discarded, which results in a set of six attention graphs with their maximum spanning trees (MSTs) becoming a chain (Figure 1c; \( \tau = 0.1 \)), and a star (Figure 1d; \( \tau = 0.5 \)). The families of attention graphs are used to compute persistent features (§3.2).

![Figure 1: An example of attention maps, barcodes, and filtration procedure for the sentence “There is snowing today.”. Model=En-BERT-base (Devlin et al., 2019). Heads=[Layer 1; Head 11] and [Layer 9; Head 9].](image)

### 3.2 Persistent Features of Attention Graphs

We follow Kushnareva et al. (2021) to design three groups of persistent features of the attention graph: (i) **topological features**, (ii) **features derived from barcodes**, and (iii) **features based on distance to attention patterns**. The features are computed on attention maps produced by a Transformer LM.

**Topological Features.** Topological features include the first two Betti numbers of the undirected graph \( \beta_0 \) and \( \beta_1 \) and standard properties of the directed graph, such as the number of strongly connected components, edges, and cycles. The features are calculated on pre-defined thresholds over undirected and directed attention graphs from each head separately and further concatenated.

**Features Derived from Barcodes.** Barcode is the representation of the graph’s persistent homology (Barannikov, 2021). We use the Ripser++ toolkit (Zhang et al., 2020) to compute 0/1-dimensional barcodes for \( A^{\text{attn}} \). Since Ripser++ leverages upon distance matrices, we transform \( A^{\text{attn}} \) as \( A' = 1 - \max (A^{\text{attn}}, A^{\text{attn}} T) \).

Next, we compute descriptive characteristics of each barcode, such as the sum/average/variance of lengths of bars, the number of bars with the time of birth/death greater/lower than a threshold, and the
entropy of the barcodes. The sum of lengths of bars \((H_0S)\) corresponds to the 0-dimensional barcode and represents the sum of edge weights in the \(A^\prime\)’s minimum spanning tree. The average length of bars \((H_0M)\) corresponds to the mean edge weight in this tree, i.e. \(H_0M = 1 - \lambda\) (the mean edge weight of the maximum spanning tree in \(A^\text{str}\)).

**Features Based on Distance to Patterns.** The shape of attention graphs can be divided into several patterns: attention to the previous/current/next token, attention to the \([\text{SEP}]/[\text{CLS}]\) token, and attention to punctuation marks (Clark et al., 2019). We formalize attention patterns by binary matrices and calculate distances to them as follows. We take the Frobenius norm of the difference between the matrices normalized by the sum of their norms. The distances to patterns are used as a feature vector.

**Notations.** We summarize the notations used throughout the paper:

- \(H_i\): \(i\)-th Homology Group
- \(\beta_i\): Betti number, dimension of \(H_i\)
- \(H_0S\): The sum of lengths of bars
- \(H_0M\): The average of lengths of bars
- \(PCA\): Principal Component Analysis
- \(PC^k\): Subset \(\{i\}\) of principal components
- \(\text{MST}\): Maximum Spanning Tree
- \(\text{RTD}\): Representation Topology Divergence

**3.3 Representation Topology Divergence**

Representation Topology Divergence (RTD; Barannikov et al., 2022) measures topological dissimilarity between a pair of weighted graphs with one-to-one vertex correspondence. Figure 2 outlines computation of RTD for a sentence pair in Example (2).

First, we compute attention maps for the input sentences \(S_a\) and \(S_b\) with a Transformer LM, and represent them as the weighted graphs \(G_a\) and \(G_b\). Next, we establish a one-to-one match between the vertices, and sort the filtrations \(G_a^\alpha\) and \(G_b^\alpha\) with \(\alpha = 1 - \tau\) in the ascending order. We then track the hierarchical formation of connected components in the graph \(G_a^\alpha \cap G_b^\alpha\) while increasing \(\alpha\). The RTD\((G_a, G_b)\) feature appears at threshold \(\alpha_i\) if an edge with the weight \(\alpha_j\) in the graph \(G_b\) joins two different connected components of the graph \(G_a^\alpha \cap G_b^\alpha\). This feature disappears at the threshold \(\alpha_j\) if the two \(G_a^\alpha \cap G_b^\alpha\) connected components become joined in the graph \(G_a^\alpha\).

**Example.** We can identify the “birth” of the RTD feature at \(\alpha = 0.47\), when an edge appears in \(G_a^\alpha = 0.47\) between the connected component “trained/murmured” and the connected component with four vertices, namely “[SEP], “[CLS]”, “.”, and “Tara” (Figure 2; the appearing edge is colored in green). We observe its “death”, when the edge becomes present in both attention graphs at \(\alpha = 0.65\) (the corresponding edge changes its color to grey in the graph \(G_a^\alpha = 0.65 \cap G_b^\alpha = 0.65\)). When comparing the graphs in this manner, we can associate a lifetime to the feature by computing the difference between the moments of its “death” (e.g., \(\alpha_i = 0.65\)) and “birth” (e.g., \(\alpha_i = 0.47\)). The lifetimes are illustrated as the orange bars \([\alpha_i, \alpha_j]\) in Figure 2. The resulting value of RTD\((G_a, G_b)\) is the sum of lifetimes \(\alpha_j - \alpha_i\) over all such features. A formal description of RTD is provided in Appendix A.
4 Acceptability Classification

4.1 Data

We use three LA classification benchmarks in English (CoLA; Warstadt et al., 2019), Italian (ItaCoLA; Trotta et al., 2021) and Swedish (DaLAJ; Volodina et al., 2021). CoLA and ItaCoLA contain sentences from linguistic textbooks and cover morphological, syntactic, and semantic phenomena. The target labels are the original authors’ acceptability judgments. DaLAJ includes L2-written sentences with morphological violations or incorrect word choices. The benchmark statistics are described in Table 1 (see Appendix C). We provide examples of acceptable and unacceptable sentences in English (3), Italian (4), and Swedish (5) from the original papers.

(3) a. What did Betsy paint a picture of?
b. *Maryann should leaving.

(4) a. Ho voglia di salutare Maria.
   “I want to greet Maria.”
b. *Questa donna mi hanno colpito.
   “This woman have impressed me.”

   “I felt very strange.”
b. *Alla blir busiga med sociala medier.
   “Everyone is busy with social media.”

4.2 Models

We run the experiments on the following Transformer LMs: En-BERT-base (Devlin et al., 2019), It-BERT-base (Schweter, 2020), Sw-BERT-base (Malmsten et al., 2020), and XLM-R-base (Conneau et al., 2020). Each LM has two instances: frozen (a pre-trained model with frozen weights), and fine-tuned (a model fine-tuned for LA classification in the corresponding language).

Baselines. We use the fine-tuned LMs, and a linear layer trained over the pooler output from the frozen LMs as baselines.

Our Models. We train Logistic Regression classifiers over the persistent features computed with each model instance: (i) the average length of bars (H0M); (ii) concatenation of all topological features referred to as TDA (§3.2). Following Warstadt et al., we evaluate the performance with the accuracy score (Acc.) and Matthew’s Correlation Coefficient (MCC; Matthews, 1975). The fine-tuning details are provided in Appendix B.

<table>
<thead>
<tr>
<th>Model</th>
<th>Frozen LMs</th>
<th>Fine-tuned LMs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>IDD / Dev</td>
<td>OODD / Test</td>
</tr>
<tr>
<td></td>
<td>Acc. MCC</td>
<td>Acc. MCC</td>
</tr>
<tr>
<td>CoLA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>En-BERT</td>
<td>69.6 0.037</td>
<td>69.0 0.082</td>
</tr>
<tr>
<td>En-BERT + HOM</td>
<td>75.0 0.338</td>
<td>75.2 0.372</td>
</tr>
<tr>
<td>En-BERT + TDA</td>
<td>77.3 0.420</td>
<td>76.7 0.420</td>
</tr>
<tr>
<td>XLM-R</td>
<td>68.9 0.041</td>
<td>68.6 0.072</td>
</tr>
<tr>
<td>XLM-R + HOM</td>
<td>71.3 0.209</td>
<td>69.8 0.187</td>
</tr>
<tr>
<td>XLM-R + TDA</td>
<td>73.0 0.336</td>
<td>70.3 0.297</td>
</tr>
<tr>
<td>ItaCoLA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>It-BERT</td>
<td>81.1 0.032</td>
<td>82.1 0.140</td>
</tr>
<tr>
<td>It-BERT + HOM</td>
<td>85.0 0.124</td>
<td>83.6 0.055</td>
</tr>
<tr>
<td>It-BERT + TDA</td>
<td>89.2 0.478</td>
<td>85.8 0.350</td>
</tr>
<tr>
<td>XLM-R</td>
<td>85.4 0.000</td>
<td>84.2 0.000</td>
</tr>
<tr>
<td>XLM-R + HOM</td>
<td>84.7 0.095</td>
<td>83.3 0.072</td>
</tr>
<tr>
<td>XLM-R + TDA</td>
<td>88.3 0.411</td>
<td>84.4 0.208</td>
</tr>
<tr>
<td>DaLAJ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sw-BERT</td>
<td>58.3 0.183</td>
<td>59.0 0.188</td>
</tr>
<tr>
<td>Sw-BERT + HOM</td>
<td>69.3 0.387</td>
<td>58.4 0.169</td>
</tr>
<tr>
<td>Sw-BERT + TDA</td>
<td>62.1 0.243</td>
<td>64.4 0.289</td>
</tr>
<tr>
<td>XLM-R</td>
<td>52.5 0.069</td>
<td>51.3 0.038</td>
</tr>
<tr>
<td>XLM-R + HOM</td>
<td>61.1 0.224</td>
<td>61.8 0.237</td>
</tr>
<tr>
<td>XLM-R + TDA</td>
<td>51.1 0.227</td>
<td>54.1 0.218</td>
</tr>
</tbody>
</table>

Table 1: Acceptability classification results by benchmark. IDD=“in domain dev” set (CoLA), OODD=“out of domain dev” set (CoLA), Dev; Test=dev and test sets in ItaCoLA and DaLAJ. The best score is put in bold, the second best score is underlined.

4.3 Results

Table 1 outlines the LA classification results. Our TDA classifiers generally outperform the baselines by up to 0.14 MCC for English, 0.24 MCC for Italian, and 0.08 MCC for Swedish. The H0M feature can solely enhance the performance for English and Italian up to 0.1, and concatenation of all features receives the best scores. Comparing the results under the frozen and fine-tuned settings, we draw the following conclusions. The TDA features significantly improve the baseline performance but require the LM to be fine-tuned for maximizing the performance. However, the TDA/H0M classifiers perform on par with the fine-tuned baselines for Swedish. The results suggest that our features may fail to infer lexical items and word derivation violations peculiar to the DaLAJ benchmark.

Effect of Freezing Layers. Another finding is that freezing the Transformer layers significantly affects acceptability classification. Most of the frozen baselines score less than 0.1 MCC across all languages. The results align with Lee et al. (2019), who discuss the performance degradation of BERT-based models depending upon the number of frozen layers. With all layers frozen, the model performance can fall to zero.
Results by Linguistic Features. We run a diagnostic evaluation of the fine-tuned models using a grammatically annotated version of the CoLA development set (Warstadt and Bowman, 2019). Figure 3 (En-BERT and XLM-R; Figure 1 in Appendix C.1) present the results of measuring the MCC of the sentences including the major features.

The overall pattern is that the TDA classifiers may outperform the fine-tuned baselines, while the $H_0M$ ones perform on par with the latter. The performance is high on sentences with default syntax (Simple) and marked argument structure, including prepositional phrase arguments (Arg. Type), and verb phrases with unusual structures (Arg. Altern). The TDA features capture surface properties, such as presence of auxiliary or modal verbs (Auxiliary), and structural ones, e.g., embedded complement clauses (Comp Clause) and infinitive constructions (to-VP). The models receive moderate MCC scores on sentences with question-like properties (Question), adjuncts performing semantic functions (Adjunct), negative polarity items, and comparative constructions (Determiner).

Analysis of the Feature Space. The LA classification experiments are conducted in the sparse feature space, where the TDA features can strongly correlate with one another, and their contribution is unclear. We run a complementary experiment to understand better how linguistic features are modeled with topology. We investigate the feature space with dimensionality reduction (principal component analysis, PCA; Pearson, 1901) by interpreting components’ structure and identifying the feature importance to the classifier’s predictions using Shapley values (Shapley, 1953), a game-theoretic approach to the attribution problem (Sundararajan and Najmi, 2020). Appendix C.2 describes the experiment on the fine-tuned En-BERT + TDA model using the grammatically annotated CoLA development set.

The results show that (i) features of the higher-layer heads, such as the average vertex degree, the number of connected components, edges, and cycles, and attention to the current token, contribute most to the major linguistic features. (ii) Attention to the [CLS] /next token is important to the Determiner, Arg. Type, Comp Clause, and to-VP properties, while attention to the first token and punctuation marks has the least effect in general. (iii) The number of nodes influences the classifier behavior, which is in line with Warstadt and Bowman, who discuss the effect of the sentence length on the performance.

5 Linguistic Minimal Pairs

5.1 Data

BLtMP (Benchmark of Linguistic Minimal Pairs; Warstadt et al., 2020a) evaluates the sensitivity of LMs to acceptability contrasts in terms of a forced choice between minimal pairs, as in Example (6). The benchmark consists of 67 pair types, each including 1k pairs covering 12 language phenomena in morphology, syntax, and semantics.

\[(6) \quad \text{a. Whose hat should Tonya wear? } \]
\[\text{b. *Whose should Tonya wear hat?}\]

5.2 Models

We conduct the experiments using two Transformer LMs for English: BERT-base and RoBERTa-base (Liu et al., 2019).

Baselines. We compare our methods with the results on BLtMP for human annotators and nine LMs (Warstadt et al., 2020a; Salazar et al., 2020). The baselines range from statistical N-gram LMs to Transformer LMs.

Our Models. Given a minimal pair as in Example (2), we build attention graphs $G_a$ and $G_b$ from each attention head of a frozen Transformer LM. We use the $H_0M$ feature (§3.2) and RTD (§3.3) as scoring functions to distinguish between the sentences $S_a$ and $S_b$. The scoring is based on empirically defined decision rules modeled after the forced-choice task:
• $H_0M$ scoring: $S_a$ is acceptable if and only if $H_0M(G_a) < H_0M(G_b)$; otherwise $S_b$ is acceptable.
• RTD scoring: $S_a$ is acceptable if and only if $\text{RTD}(G_a, G_b) < \text{RTD}(G_b, G_a)$; otherwise $S_b$ is acceptable.

We evaluate the scoring performance of each attention head, head ensembles, and all heads w.r.t. each and all linguistic phenomena in BL1IMP. The following head configurations are used for each Transformer LM and scoring function:
• Phenomenon Head and Top Head are the best-performing attention heads for each and all phenomena, respectively. The heads undergo the selection with a brute force search and operate as independent scorers.
• Head Ensemble is a group of the best-performing attention heads selected with beam search. The size of the group is always odd. We collect majority vote scores from attention heads in the group.
• All Heads involves majority vote scoring with all 144 heads. We use random guessing in case of equality of votes. This setup serves as a proxy for the efficiency of the head selection.

Notes on Head Selection. Recall that the head selection procedure\(^3\) imposes the following limitation. Auxiliary labeled minimal pairs are required to find the best-performing Phenomenon, Top Heads, and Head Ensembles. However, this procedure is more optimal and beneficial than All Heads since it maximizes the performance when utilizing only one or 9-to-59 heads. We also analyze the effect of the amount of auxiliary data used for the head selection on the scoring performance (§5.3). Appendix D.1 presents a more detailed description of the head selection procedure.

5.3 Results

We provide the results of scoring BL1IMP pairs in Table 2. The accuracy is the proportion of the minimal pairs in which the method prefers an acceptable sentence to an unacceptable one. We report the maximum accuracy scores for our methods across five experiment restarts. The general trends are that the best head configuration performs on par with the human baseline and achieves the highest overall performance (RoBERTa-base + RTD; Head Ensemble). RoBERTa predominantly surpasses BERT and other baselines, and topological scoring may improve on scores from both BERT and RoBERTa for particular phenomena.

Top Head Results. We find that $H_0M$/RTD scoring with only one Top Head overall outperforms majority voting of 144 heads (All Heads) by up to 10.6% and multiple baselines by up to 20.7% (5-gram, LSTM, Transformer-XL, GPT2-large). However, this head configuration performs worse than masked LM scoring (Salazar et al., 2020) for BERT-base (by 8.8%; Top Head=\{8; 0\}) and RoBERTa-base (by 4.6%; Top Head=\{11; 10\}).

Phenomenon Head Results. We observe that the $H_0M$/TDA scoring performance of Phenomenon Heads insignificantly differ for the same model. Phenomenon Heads generally receive higher scores than the corresponding Top Heads for BERT/RoBERTa (e.g., Binding: +17.1/+5.8%; Quantifiers: +17.7/+5.0%; Det.-Noun agr: +13.0/+1.7%), and perform best or second-best on Binding and Ellipsis. Their overall performance further adds up to 6.4/6.0% and is comparable with Salazar et al.. The results indicate that heads encoding the considered phenomena are distributed at the same or nearby layers, namely $[3; 6-9; 11]$ (BERT), and $[2-3; 8-11]$ (RoBERTa).

Head Ensemble Results. Table 3 describes the most optimal Head Ensembles by Transformer LM. Most heads selected under the $H_0M$ and RTD scoring functions are similar w.r.t. LMs. While the selected BERT heads are distributed across all layers, the RoBERTa ones tend to be localized at the middle-to-higher layers. Although RoBERTa utilizes smaller ensembles when delivering the best overall score, some heads contribute in both LMs, most notably at the higher layers.

Overall, the RoBERTa $H_0M$/RTD ensembles get the best results on Filler gap, Quantifiers, Island effects, NPI, and S-V agr as shown in Table 2, matching the human level and surpassing four larger LMs on all phenomena by up to 7.4% (GPT2-medium and GPT2/BERT/RoBERTa-large).

Effect of Auxiliary Data. Note that the head selection can be sensitive to the number of additional examples. The analysis of this effect is presented in

---

\(^3\)We acknowledge that this setup may contradict the original task formulation, that is, evaluation of pre-trained LMs on minimal pairs in the unsupervised setting (Warstadt et al., 2020a). We use auxiliary data to find the most contributing heads and analyze their linguistic roles via TDA without any model parameter updates. The generated data is publicly available for reproducibility purposes.
Table 2: Percentage accuracy of the baseline models, human baseline, and our methods on BLIMp. Overall is the average across all phenomena. The best score is put in bold, the second best score is underlined.

<table>
<thead>
<tr>
<th>Model</th>
<th>Overall</th>
<th>Ana. agr</th>
<th>Arg. agr</th>
<th>Binding</th>
<th>Cnt/Unks</th>
<th>D-S agr*</th>
<th>Ellipsis</th>
<th>Filter</th>
<th>Yr. agr</th>
<th>Island</th>
<th>NPI</th>
<th>Quant.</th>
<th>S-V agr</th>
</tr>
</thead>
<tbody>
<tr>
<td>Warstadt et al. (2020a)</td>
<td>61.2</td>
<td>47.9</td>
<td>71.9</td>
<td>64.4</td>
<td>68.5</td>
<td>70.0</td>
<td>36.9</td>
<td>60.2</td>
<td>79.5</td>
<td>57.2</td>
<td>45.5</td>
<td>53.5</td>
<td>60.3</td>
</tr>
<tr>
<td>LSTM</td>
<td>69.8</td>
<td>91.7</td>
<td>73.2</td>
<td>73.5</td>
<td>67.0</td>
<td>85.4</td>
<td>67.6</td>
<td>73.9</td>
<td>89.1</td>
<td>46.6</td>
<td>51.7</td>
<td>64.5</td>
<td>80.1</td>
</tr>
<tr>
<td>Transformer-XL</td>
<td>69.6</td>
<td>94.1</td>
<td>72.2</td>
<td>74.7</td>
<td>71.5</td>
<td>83.0</td>
<td>77.2</td>
<td>66.6</td>
<td>78.2</td>
<td>48.4</td>
<td>55.2</td>
<td>69.3</td>
<td>76.0</td>
</tr>
<tr>
<td>GPT2-large</td>
<td>81.5</td>
<td>99.6</td>
<td>78.3</td>
<td>80.1</td>
<td>80.5</td>
<td>93.3</td>
<td>86.6</td>
<td>83.3</td>
<td>79.6</td>
<td>78.9</td>
<td>71.3</td>
<td>89.0</td>
<td>71.9</td>
</tr>
<tr>
<td>Human Baseline</td>
<td>85.6</td>
<td>97.5</td>
<td>90.0</td>
<td>87.3</td>
<td>83.9</td>
<td>92.2</td>
<td>85.0</td>
<td>86.7</td>
<td>94.9</td>
<td>89.8</td>
<td>88.1</td>
<td>86.6</td>
<td>90.9</td>
</tr>
</tbody>
</table>

Salazar et al. (2020)

<table>
<thead>
<tr>
<th>Model</th>
<th>Overall</th>
<th>Ana. agr</th>
<th>Arg. agr</th>
<th>Binding</th>
<th>Cnt/Unks</th>
<th>D-S agr*</th>
<th>Ellipsis</th>
<th>Filter</th>
<th>Yr. agr</th>
<th>Island</th>
<th>NPI</th>
<th>Quant.</th>
<th>S-V agr</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPT2-medium</td>
<td>82.6</td>
<td>99.4</td>
<td>83.4</td>
<td>77.8</td>
<td>83.0</td>
<td>96.3</td>
<td>86.3</td>
<td>81.3</td>
<td>94.9</td>
<td>71.7</td>
<td>74.7</td>
<td>74.1</td>
<td>88.3</td>
</tr>
<tr>
<td>BERT-base</td>
<td>84.2</td>
<td>97.0</td>
<td>80.0</td>
<td>82.3</td>
<td>79.6</td>
<td>97.6</td>
<td>89.4</td>
<td>83.1</td>
<td>96.5</td>
<td>73.6</td>
<td>84.7</td>
<td>71.2</td>
<td>92.4</td>
</tr>
<tr>
<td>BERT-large</td>
<td>84.8</td>
<td>97.2</td>
<td>80.7</td>
<td>82.0</td>
<td>82.7</td>
<td>97.6</td>
<td>86.4</td>
<td>84.3</td>
<td>92.8</td>
<td>77.0</td>
<td>83.4</td>
<td>72.8</td>
<td>91.9</td>
</tr>
<tr>
<td>RoBERTa-base</td>
<td>85.4</td>
<td>97.3</td>
<td>83.5</td>
<td>77.8</td>
<td>81.9</td>
<td>97.0</td>
<td>91.4</td>
<td>90.1</td>
<td>96.2</td>
<td>80.7</td>
<td>81.0</td>
<td>69.8</td>
<td>91.9</td>
</tr>
<tr>
<td>RoBERTa-base + RTD</td>
<td>86.5</td>
<td>97.8</td>
<td>84.6</td>
<td>79.1</td>
<td>84.1</td>
<td>96.8</td>
<td>90.8</td>
<td>88.9</td>
<td>90.6</td>
<td>83.4</td>
<td>85.5</td>
<td>70.2</td>
<td>91.4</td>
</tr>
</tbody>
</table>

Table 3: Results of selecting the best-performing Head Ensembles with HoM/RTD-based scoring. HoM heads are colored in green; RTD heads are colored in yellow.

Appendix D.2. The results show that head ensembles, their size, and average performance tend to be more stable when using sufficient examples (the more, the better); however, using only one extra example can yield the performance above 80%.

6 Discussion

Topology and Acceptability. The topological properties of the attention graph represent interpretable and versatile features for judging sentence acceptability and identifying acceptability contrasts in minimal pairs. As one of such properties, the sum length of bars (Ho/S) — and its normalized version (H0/M) — have proved to be efficient for both LA approaches. This simple feature can serve as a profitable input for LA classifiers and a scoring function to discriminate between minimal pairs. Figure 4 shows an example of the H0/S sensitivity to CoLA’s question-like properties, such as wh-movement out of syntactic islands and matrix and embedded questions. We provide more examples in Appendix E, which demonstrate the distribution shifts between the acceptable and unacceptable sentences.

Acceptability Phenomena. The underlying structure of the attention graph encodes various well-established grammatical concepts. We observe that the persistent graph features capture surface properties, morphological agreement, structural relationships, and simple/complex syntactic phenomena well. However, with topology, lexical items, optional syntactic elements, and abstract semantic factors may be difficult to infer. Attention
7 Conclusion and Future Work

Our paper studies the ability of attention heads to judge grammatical acceptability, demonstrating the profitable application of TDA tools to two LA paradigms. Topological features can boost LA classification performance in three typologically close languages. The $H_{0M}$/RTD scoring matches or outperforms larger Transformer LMs and reaches human-level performance on BL1MP, while utilizing 9-to-59 attention heads. We also interpret the correspondence between the persistent features of the attention graph and grammatical concepts, revealing that the former efficiently infer morphological, structural, and syntactic phenomena but may lack lexical and semantic information.

In our future work we hope to assess the linguistic competence of Transformer LMs on related resources for typologically diverse languages and analyze which language-specific phenomena are and are not captured by the topological features. We are also planning to examine novel features, e.g., the number of vertex covers, the graph clique-width, and the features of path homology (Grigor’yan et al., 2020). Another direction is to evaluate the benefits and limitations of the $H_{0,M}/RTD$ features as scoring functions in downstream applications.

We also plan to introduce support for new deep learning frameworks such as MindSpore

4https://www.mindspore.cn/

8 Limitations

8.1 Computational complexity

Acceptability classification. Calculation of any topological feature relies on the Transformer’s attention matrices. Hence, the computational complexity of our features is not lower than producing an attention matrix with one head, which is asymptotically $O(n^2 d + nd^2)$ given that $n$ is the maximum number of tokens, and $d$ is the token embedding dimension (Vaswani et al., 2017).

The calculation complexity of the pattern-based and threshold-based features is done in linear time $O(e + n)$, where $e$ is the number of edges in the attention graph. In turn, the number of edges is not higher than $n(n-1)/2 \sim n^2$. The computation of the $0^{th}$ Betti number $\beta_0$ takes linear time $O(e+n)$, as $\beta_0$ is equal to the number of the connected components in an undirected graph. The computation of the $1^{st}$ Betti number $\beta_1$ takes constant time, since $\beta_1 = e - n + \beta_0$. The computational complexity of the number of simple cycles and the 1-dimensional barcode features is exponential in the worst case. To reduce the computational burden, we stop searching for simple cycles after a pre-defined amount of them is found.

Note that the computational costs could be reduced, e.g., by identifying the most contributing features or the best-performing heads. Consider an example in Figure 5, which illustrates how the CoLA performance gets changed depending on the number of En-BERT heads. Here, the head selection is based on a simple procedure. First, we score the attention heads by calculating the maximum correlation between each head’s features and the vector of the target classes on the train set. Second, we train a linear classifier over the TDA features produced by $N$ attention heads ranked by the correlation values, as specified in §4.2.
MCC scores can be achieved when utilizing less than 40 heads with a significant speed up at the inference stage.

**Linguistic Minimal Pairs.** Computation of the $H_0 M$ and RTD features is run via the Ripser++ GPU library. Under this library, the minimum spanning tree is found according to Kruskal’s algorithm giving the computational complexity of $H_0 M$ as $O(n^2 \log n)$. The complexity can be reduced using other algorithms, e.g., Prim’s algorithm, which takes $O(n^2)$. The RTD’s computational complexity is more difficult to estimate. RTD is computed via persistence barcodes of dimension 1 for a specific graph with $2n$ vertices. Many optimization techniques and heuristics are implemented in the Ripser++ library that significantly reduce the RTD’s complexity.

**Empirical estimate.** Computing the $H_0 M/RTD$ features with 144 BERT heads in the worst case of a 512-token text takes 2.41 and 94.5 sec (NVIDIA Tesla K80 12GB RAM). However, the actual computation time on the considered tasks is empirically more optimal. We provide empirical estimates on the entire BLiMP and LA datasets: 2.4/15.7 hours on BLiMP ($H_0 M/RTD$) and up to 2 hours on COLOA/ItaCoLOA/DALAJ (estimates by the feature groups: topological features=24%; features derived from barcodes=70%; and features based on distance to patterns=6% of the total time).

### 8.2 Application Limitations

We also outline several application limitations of our approach. (i) The LA classifiers require preliminary fine-tuning of Transformer LMs to extract more representative attention graph features and, therefore, achieve better performance. (ii) RTD operates upon a one-to-one vertex correspondence, which may be hindered by tokens segmented into an unequal amount of sub-tokens. As a result, identifying the topological discrepancy between pairs of attention graphs can be restricted in practice, where the graphs are of an arbitrary number of nodes. Regardless of the potential information loss due to sentence truncation in such cases, the RTD heads still receive the best overall score on BLiMP. (iii) The head selection procedure relies on auxiliary data to identify the best-performing head configurations. Annotating the auxiliary data may require additional resources and expertise for practical purposes. However, the procedure maximizes the performance and reduces the computational costs by utilizing less attention heads.

### 8.3 Linguistic Acceptability

Acceptability judgments have been broadly used to investigate whether LMs learn grammatical concepts central to human linguistic competence. However, this approach has several methodological limitations. (i) The judgments may display low reproducibility in multiple languages (Linzen and Oseki, 2018), and (ii) be influenced by an individual’s exposure to ungrammatical language use (Dąbrowska, 2010). (iii) Distribution shifts between LMs’ pre-training corpora and LA datasets may introduce bias in the evaluation since LMs tend to assign higher probabilities to frequent patterns and treat them as acceptable in contrast to rare ones (Marvin and Linzen, 2018; Linzen and Baroni, 2021).

### 9 Ethical Statement

Advancing acceptability evaluation methods can improve the quality of natural language generation (Batra et al., 2021). We recognize that this, in turn, can increase the misuse potential of such models, e.g., generating fake product reviews, social media posts, and other targeted manipulation (Jawahar et al., 2020; Weidinger et al., 2021). However, the acceptability classifiers and scoring functions laid out in this paper are developed for research purposes only. Recall that the topological tools can be employed to develop adversarial defense and artificial text detection models for mitigating the risks (Kushnareva et al., 2021).
Acknowledgements

This work was supported by Ministry of Science and Higher Education grant No. 075-10-2021-068 and by the Mindspore community. Irina Proskurina was supported by the framework of the HSE University Basic Research Program.

References


A  

Representation Topology Divergence

Suppose we have two weighted full graphs $G_a, G_b$ with one-to-one vertex correspondence. Define their vertices as $\{a_1, a_2, \ldots, a_n\}$ and $\{b_1, b_2, \ldots, b_n\}$ respectively so that $a_i$ corresponds to $b_i$ for each $i$. RTD($G_a, G_b$) is calculated as follows:

1. Build a full weighted graph $G_{ab}$ with the vertices set $V = \{v_1, v_2, \ldots, v_n, u_1, u_2, \ldots, u_n\}$ and the edge weights computed as

   \[
   \begin{align*}
   w(v_i, v_j) &= 0 \\
   w(v_i, u_j) &= 0 \\
   w(u_i, u_j) &= w_b(b_i, b_j) \\
   w(v_i, u_j) &= \max(w_a(a_i, a_j), w_b(b_i, b_j))
   \end{align*}
   \]

   where $w_a$ and $w_b$ are the edge weights in the corresponding graphs.

2. Compute the barcode (Barannikov, 2021) of the $H_1$ homology group of the graph $G_{ab}$ flag complex. It should be emphasized that the $H_0$ homology group barcode for this graph is empty since the minimum spanning tree of $G_{ab}$ has the total weight of 0. Instead of $H_1$, the higher-order homology groups (e.g., $H_2, H_3$) can be considered. However, the preliminary experiments have shown that they are less helpful for LA tasks.

3. RTD($G_a, G_b$) is calculated as the sum of bar lengths in the barcode from the previous step.

   It should be noted that this procedure is asymmetric on $G_a$ and $G_b$, and for non-equal graphs holds $\text{RTD}(G_a, G_b) \neq \text{RTD}(G_b, G_a)$. To compute barcodes, we use the Ripser++ toolkit, which cannot work with asymmetric graphs. Hence, we represent the asymmetric attention maps as the distance matrices to obtain the symmetric graphs $G_a$ and $G_b$ as described in §3.2. We consider only the forward-looking part of attention, i.e., how each token affects the rest of the sentence.

   The majority of the BLIMP minimal pairs are of equal length in the BERT/RoBERTa tokens. Otherwise, we truncate the longest sentence to achieve an equal length since the one-to-one correspondence between tokens is crucial for RTD. We assume that the truncation may remove tokens that help to discriminate between the acceptable and unacceptable sentences. We leave improvement of the pre-processing stage for future work.

B  

Fine-tuning Details

Fine-tuning and evaluation of the BERT-based/XLM-R acceptability classifiers follow the standard procedure under the HuggingFace library (Wolf et al., 2020). Each model is fine-tuned for 4 epochs with the learning rate of $1e^{-2}/1e^{-3}$, batch size of 32, and the other default hyperparameters.

C  

Acceptability Classification

<table>
<thead>
<tr>
<th></th>
<th>CoLA</th>
<th>ItaCoLA</th>
<th>DaLAJ</th>
</tr>
</thead>
<tbody>
<tr>
<td>Language</td>
<td>English</td>
<td>Italian</td>
<td>Swedish</td>
</tr>
<tr>
<td># Train sent.</td>
<td>8,551</td>
<td>7,801</td>
<td>6,870</td>
</tr>
<tr>
<td># Dev sent.</td>
<td>1,043</td>
<td>946</td>
<td>892</td>
</tr>
<tr>
<td># Test sent.</td>
<td>1,063</td>
<td>975</td>
<td>952</td>
</tr>
<tr>
<td>Type</td>
<td>Expert</td>
<td>Expert</td>
<td>L2</td>
</tr>
<tr>
<td># Sources</td>
<td>23</td>
<td>12</td>
<td>SweL1</td>
</tr>
<tr>
<td>Phenomena</td>
<td>Morph, Syntax, Semantics</td>
<td>Syntax, Lexis, Morph</td>
<td></td>
</tr>
<tr>
<td>%</td>
<td>70.5</td>
<td>84.5</td>
<td>50.0</td>
</tr>
</tbody>
</table>

Table 1: Statistics of acceptability classification benchmarks. Type=Type of data source. %=Percentage of acceptable sentences. Morph=Morphology.

C.1  

Results by Linguistic Features

![Figure 1: Performance (MCC) of the fine-tuned XLM-R by major linguistic feature. Average MCC scores are represented with dashed lines. The number of sentences including the feature is placed in square brackets.](image)

C.2  

Analysis of the Feature Space

We analyze the contribution of the topological features to acceptability classification in the context of linguistic phenomena. We interpret the principal components computed on the fine-tuned En-BERT + TDA features and identify their importance w.r.t. each head/layer with Shapley values.
Table 2: Acceptability classification results with PCA on CoLA. IDD="in domain dev" set. OODD="out of domain dev" set. Components Set 1: {1}, Components Set 2: {1,7,9,24,12,0}.

<table>
<thead>
<tr>
<th>Model</th>
<th>IDD Acc.</th>
<th>MCC</th>
<th>OODD Acc.</th>
<th>MCC</th>
</tr>
</thead>
<tbody>
<tr>
<td>En-BERT + TDA</td>
<td>88.6</td>
<td>0.725</td>
<td>82.1</td>
<td>0.565</td>
</tr>
<tr>
<td>En-BERT + TDA + PCA</td>
<td>84.8</td>
<td>0.632</td>
<td>81.8</td>
<td>0.558</td>
</tr>
<tr>
<td>En-BERT + TDA + PC(^1)</td>
<td>84.1</td>
<td>0.609</td>
<td>79.1</td>
<td>0.482</td>
</tr>
<tr>
<td>En-BERT + TDA + PC(^2)</td>
<td>84.3</td>
<td>0.615</td>
<td>81.2</td>
<td>0.541</td>
</tr>
</tbody>
</table>

**Method.** The pipeline assembles the feature standardization, PCA and training a logistic regression classifier. We conduct a grid search over two pipeline’s parameters: (i) the number of components \(N_{\text{comp}} \in [10, 20, \ldots, 100]\) (the found optimum: \(N_{\text{comp}} = 100\)) and (ii) regularization parameter of logistic regression \(L_1 \in [0.01, 0.02, \ldots, 0.1]\) (the found optimum: \(L_1 = 0.1\)). The parameter search is run across 3 stratified folds, where the CoLA train set is randomly split into train/development sets. The classifier performance is evaluated on the grammatically annotated CoLA development set. We also explore masking principal components, i.e., training the classifier using only the most important components while zeroing the weights of the others.

Table 2 shows results for the full pipeline (En-BERT + TDA + PCA) and masked pipelines (En-BERT + TDA + PC\(^1\)/PC\(^2\)). Since the performance is comparable with the En-BERT + TDA classifier in §4, we rely on the PCA decomposition for the feature analysis and interpretation.

**Results.** The following six principal components (PCs) contribute most to acceptability classification according to the mean absolute Shapley values \(\phi\) (see Figure 2). Figure 3 shows the Shapley values for these PCs by the major linguistic feature.

- **PC1** (\(\phi=0.179\)) has the most impact on the classifiers’ performance. **PC1** primarily contains simple topological features (the average vertex degree, the number of edges, and the number of connected components) from the heads at the last layer, which is affected most by the fine-tuning.
- **PC7** (\(\phi=0.601\)) includes same heads as **PC1**, but its features utilize the number of cycles in the attention graph.
- **PC9** (\(\phi=0.442\)) groups all attention patterns except for attention to commas for heads at the lower and middle layers. The component attributes to all phenomena.
- **PC24** (\(\phi=0.296\)) is responsible for topological features at the first and last layers.
- **PC12** (\(\phi=0.278\)) contains the attention to the \([\text{CLS}]\) token/next token patterns. The PC is important for classifying sentences including the following features: negative polarity and free choice items (Determiner), obliques, expletives, prepositional phrases and arguments (Argument Type), complement clauses without complementizers (Complement Clause).
- **PC0** (\(\phi=0.274\)) represents features equal to the number of nodes in the graph, that is, the sentence length in tokens. The PC influences the classifier prediction w.r.t. most of the phenomena, except for Determiner, Complement Clause, and Argument Types.

The following four PCs are less important for acceptability classification (\(\phi_j < 0.25\)) in general but may contribute to some linguistic phenomena.

- **PC16** (\(\phi=0.243\)) comprises topological and distance to pattern features of different heads at the middle layers. The PC contributes to negative polarity and free choice items, non-finite complementizer phrases, and comparative constructions.
- **PC20** (\(\phi=0.216\)) reflects attention-to-comma for various heads at the lower layers. However, this feature helps to classify sentences that fall under the S-Syntax and Question categories.
mean(|SHAP value|) (average impact on model output magnitude)

We analyze the effect of the amount of auxiliary minimal pairs, the

Figure 3: Concatenated mean absolute Shapley values
for the important PCs by major linguistic feature.

PC15 ($\phi=0.216$) includes the attention to the first

PC2 ($\phi=0.203$) reflects the number of graph edges
for heads at the first layer, which captures strong

PC3 and PC6 ($\phi < 0.05$) represent attention to the
dot pattern. The PCs are not important for
any of the linguistic phenomena and have large
eigenvalues.

D Attention Head Selection

D.1 Head Selection Procedure

We use publicly available scripts\footnote{github/alexwarstadt/data_generation} to generate up to
100 minimal pairs per each of 67 types, ensuring no overlap with the BL1MP pairs. We select the
best-performing individual heads and head ensembles by estimating their scoring performance on
the generated data and further evaluate them on BL1MP.

Algorithms 1-2 describe the Top Head and Phe-

per language phenomenon used for selecting Head
Ensembles as described in Appendix D.1. The

experiments are run ten times, where each run in-
cludes generation of auxiliary minimal pairs, the
corresponding head selection procedure, and eval-
uation on BL1MP for each and all language phe-
nomena. The accuracy performance is averaged
over all experiment runs.

\begin{algorithm}
\begin{algorithmic}
\caption{Top Head Selection}
\begin{itemize}
\item[]\textbf{Input:} Set $Q_1$: contains all possible pairs $(h, r)$, where $h$ – attention head
and $r \in \{1, 2\}$ – scoring rule.
\item[]\textbf{Require:} $acc(.)$ : accuracy evaluation function of the head with the selected
rule on pairs for all phenomena.
\item[]\textbf{Output:} Pair $(H_B, R_B)$
\item[]\textbf{procedure} SELECTING Top Head$(Q_1)$
\item[]2: \textbf{BestAcc} $\leftarrow 0$
\item[]4: \textbf{for} $(h, r) \in Q_1$ \textbf{do}
\item[]5: \textbf{if} $acc((h, r)) > \text{BestAcc}$ \textbf{then}
\item[]6: \hspace{1em} $\text{BestAcc} \leftarrow acc_C((h, r))$
\item[]8: \textbf{end if}
\item[]10: \textbf{return:} $(H_B, R_B)$
\item[]\end{algorithmic}
\end{algorithm}

\begin{algorithm}
\begin{algorithmic}
\caption{Phenomenon Head Selection}
\begin{itemize}
\item[]\textbf{Input:} Set $Q_1$: contains all possible pairs $(h, r)$, where $h$ – attention head
and $r \in \{1, 2\}$ – scoring rule.
\item[]\textbf{Require:} $acc(.)$ : accuracy evaluation function of the head with the selected
scoring rule on the $C$ pairs.
\item[]\textbf{Output:} Pair $(H_C, R_C)$
\item[]\textbf{procedure} SELECTING Phenomenon Head$(Q_1)$
\item[]2: \textbf{BestAcc} $\leftarrow 0$
\item[]4: \textbf{for} $(h, r) \in Q_1$ \textbf{do}
\item[]5: \textbf{if} $acc((h, r)) > \text{BestAcc}$ \textbf{then}
\item[]6: \hspace{1em} $\text{BestAcc} \leftarrow acc_C(h, r))$
\item[]8: \textbf{end if}
\item[]10: \textbf{return:} $(H_C, R_C)$
\item[]\end{algorithmic}
\end{algorithm}

\begin{algorithm}
\begin{algorithmic}
\caption{Head Ensemble Selection}
\begin{itemize}
\item[]\textbf{Input:} Set $Q_1$: contains all possible pairs $(h, r)$, where $h$ – attention head
and $r \in \{1, 2\}$ – scoring rule.
\item[]\textbf{Require:} $acc(.)$ : accuracy evaluation function of the ensembles with selected
scoring rules using majority voting.
\item[]\textbf{Output:} Ensemble $B$: set of pairs $(H, R)$
\item[]\textbf{procedure} SELECTING Head Ensemble$(Q_1)$
\item[]2: $Q \leftarrow \{(h, r) | (h, r) \in Q_1\}$
\item[]4: \textbf{for} $q \in Q$ \textbf{do}
\item[]6: \textbf{for} $(h_1, r_1) \neq (h_2, r_2) \in Q_1 \ \text{and} \ q \text{ do}
\item[]8: \textbf{if} $acc(q') > acc(q)$ \textbf{then}
\item[]10: \hspace{1em} $Q' \leftarrow Q' \cup \{q'\}$
\item[]12: \textbf{end for}
\item[]14: \textbf{if} $|Q'| \geq 40$ \textbf{then}
\item[]16: \textbf{if} $|Q'| > 0$ \textbf{then}
\item[]18: \hspace{1em} $Q \leftarrow Q'$
\item[]20: \textbf{end if}
\item[]22: \textbf{return:} Ensemble from $Q$
\item[]\end{algorithmic}
\end{algorithm}
Results. Figure 4 presents the results for the BERT-base and RoBERTa-base models. We observe that some phenomena receive prominent performance using only one auxiliary example (e.g., BERT-base: Irregular forms: 85%; Determiner-Noun agr: 84%; Anaphor agr: 80%; Subject-verb agr: 80%; RoBERTa-base: Anaphor agr: 90%; Determiner-Noun agr: 87%; Subject-verb agr: 83%). Both models receive similar maximum scores for some phenomena with a given different amount of examples (Control / raising: 79%/80%, N=30/60; Ellipsis: 88%/87%, N=80/90; Filler gap: 80%/88%, N=60/30). By contrast, there is a significant difference between the maximum and minimum scores on certain phenomena, e.g., Quantifiers (10%/0%), NPI licensing (8%/6%), and Ellipsis (8%/6%).

E The $H_0S$ Feature Distributions

Figure 5 and Figure 6 illustrate examples of the $H_0S$ feature distribution shifts between the acceptable and unacceptable sentences from the entire CoLA development set.

Let us demonstrate a calculation of essential barcode features of a toy graph $G_{toy}$ (Figure 7a). First, we calculate the $H_0$-barcode of this graph. To do it, we build the graph $G'_{toy}$ by replacing each edge weight $w$ with $1 - w$, as in Figure 7b. Next we calculate the minimum spanning tree of this new graph (Figure 7c). We end up with the $H_0$-barcode with the lengths of bars, equal to the weights of the minimum spanning tree (Figure 7d). We can derive $H_0S(G_{toy}) = 0.3 + 0.4 + 0.5 = 1.2$ and $H_0M(G_{toy}) = H_0S(G_{toy})/3 = 0.4$ from this barcode diagram.

Note that the directions of the bars (Figure 1) were reversed comparing to the actual ripser++ output (Figure 7d). The reversed representation is more intuitive: edges with lower weights are filtered out earlier than edges with the higher weights.

Next we compute the Betti numbers for the same graph $G_{toy}$ given three thresholds: $\tau_1 = 0$, $\tau_2 = 0.4$ and $\tau_3 = 1$. 

Figure 5: The distribution shift of the $H_0S$ feature between the acceptable and unacceptable sentences (Simple); [L: 10; H: 3].

Figure 6: The distribution shift of the $H_0S$ feature between the acceptable and unacceptable sentences ( Binding); [L: 11; H: 4].

Figure 7: The effect of a given amount of examples on the BLtMP performance of selected Head Ensembles by major category. Method=RTD scoring. N=number of extra examples per phenomenon.
At $\tau = 0$ we do not drop any edges. We have the full graph with one connected component (Figure 7a). $\beta_0$ is defined as the number of connected components. Hence $\beta_0$ equals to 1. Next we calculate $\beta_1$, using the shortcut formula for graphs: $\beta_1 = |E| + |C| - |V|$. In our case, $|E| = 6$ is the number of edges, $|C| = 1$ is the number of connected components and $|V| = 4$ is the number of vertices. Finally, we get $\beta_1 = 3$. Note that $\beta_1$ corresponds to three simple undirected loops in the graph. There is also an alternative method to represent the graph and to calculate the first Betti number. This method does not account for “trivial” loops, which are defined by the triangle borders. It was used in the example above, see Figure 1.

At $\tau = 0.4$, we drop all edges with weights lower than 0.4. We get the same structure as the minimum spanning tree of the graph $G'_{toy}$ (Figure 7c), but without weights inversion. For this graph, $\beta_0 = 1$: there is a single connected component, $\beta_1 = 3 + 1 - 4 = 0$. It corresponds to the number of simple loops, which equals to 0.

At $\tau = 1$, we drop all edges as all edges have weights below than 1. The resulting graph consists only of vertices without edges. For this case, we have four connected components, so $\beta_0 = 4$, and $\beta_1 = 0 + 4 - 4 = 0$. 

Figure 7: An example of a weighted graph and corresponding $H_0$-barcode, calculated with the Ripser++ library.