Betti numbers of attention graphs is all you really need

Anonymous Author(s)
Affiliation
Address
email

Abstract

We apply methods of topological analysis to the attention graphs, calculated on the attention heads of the BERT model (Devlin et al. (2019)). Our research shows that the classifier built upon basic persistent topological features (namely, Betti numbers) of the trained neural network can achieve classification results on par with the conventional classification method. We show the relevance of such topological text representation on three text classification benchmarks. For the best of our knowledge, it is the first attempt to analyze the topology of an attention-based neural network, widely used for Natural Language Processing.

1 Introduction

Modern Neural Networks embed data into a high-dimensional space. Moreover, each layer and even a layer part can be considered as separate embedding, where the information about interconnections of these separate embeddings is encoded by some weighted directed graph. In particular, one can apply various methods to investigate such graphs for attention heads in multi-headed attention models, such as BERT.

Conventionally, the BERT model is used for sentence classification by adding a softmax-based classification layer upon the output embedding. Instead, we propose to use a linear classifier built solely upon the persistent topological features (namely, the first two Betti numbers) without using any information about the order of tokens or to which particular token each weight relates. We have found that it provides a classification quality on par with the conventional classification method in numerous tasks. Moreover, on some tasks such as linguistic acceptability and spam detection, our topological classifiers outperform the usual BERT-based classification. We conclude that the topology of the attention graphs of the trained BERT model contains enough information for solving considered classification tasks. The second outcome is, that the proposed text representation, based only on the first two Betti numbers of the attention graph, can solve the task having lower dimensionality than BERT embedding.

The attention graphs are built as follows. Each attention head in the Transformer architecture calculates weights of each token in the sentence with respect to every other token, and the next level representation is constructed using these weights. The attention graph for each head is a complete digraph (with loops which appear when the token "pays attention" to itself) whose vertices are the tokens and the attention weights are the weights of the edges.

In the classifiers, we use the BERT-based classification model, which is initialized with pre-trained BERT weights and then is fine-tuned for a given two-class classification task. After fine-tuning, we extract the persistent features of each head of this model for each data sample and then train a logistic regression classifier upon these features.

Submitted to the Topological Data Analysis and Beyond Workshop at the 34th Conference on Neural Information Processing Systems (NeurIPS 2020). Do not distribute.

Note that our results also confirm that different attention heads contain different amounts of information. These results are well aligned to previous works on BERT (Michel, Levy, and Neubig (2019), Clark et al. (2019)).

2 Related work

There are several recent insights obtained by topological analysis of the neural representations of realistic datasets. The results of Naitzat, Zhitnikov, and Lim (2020) demonstrate that a deep neural network with $ReLU$ activation function tends to simplify the topology of the data from layer to layer, with the smallest Betti numbers on the output representations. Topological features are also shown to be efficient for predicting the generalization ability of the network, its efficiency and stability to adversarial examples (Corneanu, Escalera, and Martinez (2020), Corneanu et al. (2019), Rieck et al. (2018)). An overview of persistent topology methods, both practical and theoretical, with the focus on Artificial Neural Networks analysis, can be found in Otter et al. (2017), Chazal and Michel (2017). At the same time, while many researchers are focused on applications of topology to modern AI algorithms, there are efforts in the mathematical society to further expand the set of applicable methods (Bergomi et al. (2019), Chowdhury et al. (2019), Manin and Marcolli (2020)).

3 Background

3.1 Topological background

In our approach, we use the following two numerical attributes of an arbitrary graph $G = (E, V)$: the number β_0 of connected components and the number β_1 of independent cycles of G. If one considers the graph G as a simplicial complex, these numbers are equal to its Betti numbers. Note that the Betti numbers β_0 and β_1 of a graph filtration keep the whole information about the persistent homology barcodes, see Appendix A for details.

3.2 BERT model

BERT (Devlin et al. (2019)) is the pre-trained model, which achieves state of the art results for many NLP tasks. The model is based on Transformer architecture, introduced in Vaswani et al. (2017). The BERT model is pre-trained on the large amount of data with Masked Language Modelling and Next Sentence Prediction objectives. For downstream tasks the task-specific classifier is attached to the BERT output layer and the model is fine-tuned. In our experiments we use the uncased BERT-base version, which consists of 12 layers, with 12 attention heads in each. The input of each attention head is a matrix X consisting of the d-dimensional representations (row-wise) of m tokens of the sentence, so that X is of size $m \times d$. The output of the head is the updated matrix of the representations X^{out}, that is,

$$\begin{aligned} X^{\mathrm{out}} &= W^{\mathrm{attn}}(XW^{\mathrm{V}}) \\ \text{with } W^{\mathrm{attn}} &= \mathrm{softmax}\left(\frac{(XW^{\mathrm{Q}})(XW^{\mathrm{K}})^{\mathrm{T}}}{\sqrt{d}}\right), \end{aligned} \tag{1}$$

where W^{Q}, W^{K}, W^{V} are trained projection matrices of size $d \times d$ and W^{attn} is the $m \times m$ matrix of attention weights (cf. (Vaswani et al., 2017, Sec. 3.2)). One can interpret each element w_{ij}^{attn} as a weight of j-th input's influence on i-th output; larger weights mean stronger connection between corresponding tokens.

4 Our method

Let us be given some dataset $S = \{s_i\}_{i=1}^{N}$ of N natural language texts encoded with m tokens each and pre-trained attention-based model M. First of all, we fix some set of thresholds $T = \{t_i\}_{i=1}^{k}, 0 < t_1 < t_2 < ... < t_k < 1$ and chose a subset of heads of the model H_M, on which we will perform calculations.

Then we feed each text sample $s = s_i$ to the input of the model M and obtain the matrix $W^{\mathrm{attn}} = (w_{i,j}^{attn})$ on each head $h \in H_M$. This matrix defines a weighted complete digraph with loops $\Gamma_{\mathsf{s}}^{\mathsf{h}}$ with m vertices, where w_{ij}^{attn} is the weight of the edge $j \to i$.

After it, for each graph Γ_s^h and for each threshold level $t_i \in T$ we build an unweighted directed graph $\Gamma_s^h(t_i)$ as follows. The set of vertices of $\Gamma_s^h(t_i)$ is the same as the one for the graph Γ_s^h, moreover, an edge of Γ_s^h belongs to the new graph $\Gamma_s^h(t_i)$ if and only of its weight in Γ_s^h is at least t_i. This way we assign a sequence of graphs $\Gamma_s^h(t_i), t_i \in T$ to each text sample for each head of the model.

For each unweighted directed graph $\Gamma_s^h(t_i)$ we also consider the corresponding undirected graph $\overline{\Gamma_s^h(t_i)}$ by setting an undirected edge v_1v_2 for each pair of vertices v_1 and v_2 which are connected by an edge in at least one direction in the graph $\Gamma_s^h(t_i)$. Then we count β_0, β_1 of undirected graph $\overline{\Gamma_s^h(t_i)}$. More precisely the process of features calculation for each data sample is described in Algorithm 1.

Algorithm 1 Topological features calculation

Require: Text sample s
Require: Set of chosen attention heads H_M of attention-based model M
Require: Thresholds array T
Ensure: Features array $Features$

procedure FEATURES_CALCULATION(s, H_M, T)
  for all $h \in H_M$ **do**
    Calculate attention graph $\Gamma_s^h = (V, E, W_{h,s}^{attn})$ on sample s on head h
    for all $t \in T$ **do** ▷ Filtration:
      $E_s^h(t) \leftarrow \{e \in E(\Gamma_s^h) : W_{h,s}^{attn}(e) \geq t\}$ ▷ Removing edges of weight less than t
      $\Gamma_s^h(t) \leftarrow (V, E_s^h(t))$ ▷ Ignoring weights of remaining edges
      $\overline{E_s^h(t)} \leftarrow \big\{\{i,j\} : (i,j) \in E_s^h(t)\big\}$ ▷ Ignoring edges directions
      $\overline{\Gamma_s^h(t)} \leftarrow (V, \overline{E_s^h(t)})$
      Calculate $\beta_0(\overline{\Gamma_s^h(t)}), \beta_1(\overline{\Gamma_s^h(t)})$ ▷ Calculating Betti numbers of undirected graph
    end for
  end for
  $Features \leftarrow \left[\beta_0(\overline{\Gamma_s^h(t)}), \beta_1(\overline{\Gamma_s^h(t)})\right]_{t\in T}^{h\in H_M}$
  return $Features$
end procedure

After this features calculation, we train the logistic regression on features, obtained for each sample of the train subset of the dataset and then make predictions on features of samples from the test subset.

5 Experiments

5.1 Datasets

We performed our experiments on the following datasets, labeled for different classification tasks.

The Corpus of Linguistic Acceptability ("CoLA") dataset (Warstadt, Singh, and Bowman (2018)) contains 10,657 sentences, labeled by acceptability (grammaticality) and divided into public (open) and test (hidden) parts. The public part of dataset contains 9,594 sentences and is divided, in turn, into training and development ("CoLA$_{dev}$") sets. The test set ("CoLA$_{test}$") contains 1,063 sentences with labels, hidden from the developer.

Large Movie Review Dataset v1.0 ("IMDB") (Maas et al. (2011)) contains 50,000 movie reviews, labeled by sentiment: "positive" or "negative". Labeled reviews are divided into two equal subsets, purposed for training and for testing. We applied additional lengths restriction to the samples of this dataset to obtain attention graphs of a reasonable size. Namely, we kept all reviews of size less than 128 tokens after tokenization with standard BERT uncased tokenizer ("Imdb$^{\leq 128}$"), and pruned away all others. After it, 5505 reviews remained in total. Then we divided the subset, suggested for testing purposes, into equal development and test sets.

The SMS Spam Collection v.1 ("SPAM") (Almeida, Hidalgo, and Yamakami (2011)) is a public set of SMS (text) labeled messages that have been collected for mobile phone spam research. It contains

5,574 real and non-encoded messages, tagged as legitimate (ham) or spam. For our purposes, we divided it into train, development and test ("SPAM$_{test}$") sets randomly in proportion $80:10:10$.

We used "development" subsets for tuning logistic regression hyperparameters: maximum amount of iterations and l_2-regularization coefficient. "Test" subsets were used for final validation.

5.2 Results

	CoLA$_{dev}$	CoLA$_{test}$	Imdb$^{\leq 128}_{test}$	SPAM$_{test}$
BERT	0.559 (82.0%)	0.492	0.833 (91.7%)	0.941 (98.7 %)
β_0, β_1, 144 heads	0.549 (81.1 %)	**0.508**	0.812 (90.6 %)	**0.950** (**98.9** %)
β_0, β_1, 12 best heads	0.532 (80.7 %)	0.463	0.805 (90.3 %)	0.878 (97.3 %)
β_0, β_1, 3 best heads	0.452 (77.1 %)	0.456	0.799 (90.0 %)	0.809 (96.1 %)
β_0, β_1, 1 best head	0.427 (76.4 %)	0.385	0.735 (86.9%)	0.606 (92.3 %)
Test examples amount	1043	1033	1415	556

Table 1: The comparison of our classification methods with the conventional BERT-based classifier by Matthew score and accuracy (in brackets, %).

As an efficiency measure of a linear classifier, we use Matthew score (Matthew coefficient), which is calculated by formula

$$\text{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}},$$

where we denote by FP, FN, TN, and TP the amount of false positive, false negative, true negative, and true positive predictions of our classifier, respectively. We also note classifications accuracy in brackets for those datasets, where test labels are available in open access.

For these experiments we fine-tuned BERT on each of datasets separately and used the set of six weight thresholds for calculating Betti numbers. In Table 1 we emphasized in bold the results which surpassed the result of the fine-tuned BERT classifier.

For the first experiment we used the features calculated on all 144 heads. For consequent experiments, we checked Matthew score of classification upon features, built from the graph on each head on the train set, and ranged heads in descending order according to it. Then we picked 12, 3 or 1 heads with the best Matthew score and used them for calculation of classification features (Betti numbers) on development and test sets.

It's noticeable that topological features of attention graphs on particular heads have different linear separability. For more information about this see Appendix B.

6 Conclusion and further research

We have shown that the topology of attention graphs contains enough information for classifying texts by three different attributes: linguistic acceptability, sentiment, and being SPAM or not. Thus, we see here some degree of universality for distinguishing different text properties.

Moreover, the result of our linear classifier, trained on topological features, surpassed the result of the conventional BERT-based classifier on the hidden test subset of the Corpus of Linguistic Acceptability dataset and is a little better on the SMS Spam Collection v. 1 dataset. This allows us to suppose that these features may contain even more generalized task-relevant information than the BERT output embedding. Plans of our future research include checking this daring statement with other topological features and other threshold collections. Particularly, in our current work we didn't use the information about directions of graph edges, which could be utilized with directed graph invariants, such as number of simple directed cycles and number of strongly connected components of a digraph.

Another possible direction for future work is to use the information about differences between linear separability scores on different heads to determine which heads are more or less important for each particular task. Which can potentially be used as a base for new strategies of efficiently decreasing the model size.

References

Almeida, T. A.; Hidalgo, J. M. G.; and Yamakami, A. 2011. Contributions to the study of SMS spam filtering: new collection and results. In Hardy, M. R. B.; and Tompa, F. W., eds., *ACM Symposium on Document Engineering*, 259–262. ACM. ISBN 978-1-4503-0863-2. URL http://dblp.uni-trier.de/db/conf/doceng/doceng2011.html#AlmeidaHY11.

Bergomi, M. G.; Frosini, P.; Giorgi, D.; and Quercioli, N. 2019. Towards a topological–geometrical theory of group equivariant non-expansive operators for data analysis and machine learning. In *Nature Machine Intelligence 1.9*, 423–433.

Chazal, F.; and Michel, B. 2017. An introduction to Topological Data Analysis: fundamental and practical aspects for data scientists. *ArXiv* abs/1710.04019. URL https://arxiv.org/pdf/1710.04019.pdf.

Chowdhury, S.; Gebhart, T.; Huntsma, S.; and Yutin, M. 2019. Path homologies of deep feedforward networks. In *18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, 1077–1082.

Clark, K.; Khandelwal, U.; Levy, O.; and Manning, C. D. 2019. What Does BERT Look At? An Analysis of BERT's Attention. *CoRR* abs/1906.04341. URL https://arxiv.org/pdf/1906.04341.pdf.

Corneanu, C. A.; Escalera, S.; and Martinez, A. M. 2020. Computing the Testing Error without a Testing Set. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2677–2685.

Corneanu, C. A.; Madadi, M.; Escalera, S.; and Martinez, A. M. 2019. What does it mean to learn in deep networks? And, how does one detect adversarial attacks? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4757–4766.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT (1)*.

Maas, A. L.; Daly, R. E.; Pham, P. T.; Huang, D.; Ng, A. Y.; and Potts, C. 2011. Learning Word Vectors for Sentiment Analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 142–150. Portland, Oregon, USA: Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P11-1015.

Manin, Y.; and Marcolli, M. 2020. Homotopy Theoretic and Categorical Models of Neural Information Networks. *ArXiv* abs/2006.15136. URL https://arxiv.org/pdf/2006.15136.pdf.

Michel, P.; Levy, O.; and Neubig, G. 2019. Are Sixteen Heads Really Better than One? In Wallach, H.; Larochelle, H.; Beygelzimer, A.; dÁlché Buc, F.; Fox, E.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 32*, 14014–14024. Curran Associates, Inc. URL http://papers.nips.cc/paper/9551-are-sixteen-heads-really-better-than-one.pdf.

Naitzat, G.; Zhitnikov, A.; and Lim, L.-H. 2020. Topology of deep neural networks. *ArXiv* abs/2004.06093. URL https://arxiv.org/pdf/2004.06093.pdf.

Otter, N.; Porter, M. A.; Tillmann, U.; Grindrod, P.; and Harrington, H. A. 2017. A roadmap for the computation of persistent homology. In *EPJ Data Science 6.1*, 17.

Rieck, B.; Togninalli, M.; Bock, C.; Moor, M.; Horn, M.; Gumbsch, T.; and Borgwardt, K. 2018. Neural Persistence: A Complexity Measure for Deep Neural Networks Using Algebraic Topology. In *International Conference on Learning Representations*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention Is All You Need. In *Advances in neural information processing systems*, 5998–6008.

Warstadt, A.; Singh, A.; and Bowman, S. R. 2018. Neural Network Acceptability Judgments. *arXiv preprint arXiv:1805.12471* .

Appendix A. Persistent homology and Betti numbers

Recall that a simplicial complex K is a finite collection of finite sets called *simplices* such that each subset of any element of K also is an element of K; such subsets of a simplex are called *faces*. In particular, an undirected graph is a simplicial complex where all edges and vertices are its faces. The set of all formal $\mathbb{Z}$-linear combinations of the p-dimensional simplices (that is, $(p-1)$-element simplices) of K is denoted $\mathsf{C}_p(K)$. These linear combinations $c = \sum_j \gamma_j \sigma_j$ are called p-*chains*, where the $\gamma_j \in \mathbb{Z}$ and the σ_j are p-simplices in K.

The boundary, $\partial(\sigma_j)$, is the formal sum of the $(p-1)$-dimensional faces of σ_j and the boundary of the chain is obtained by extending ∂ linearly,

$$\partial(c) = \sum_j \gamma_j \partial(\sigma_j),$$

with integer coefficients γ_j.

The p-chains that have boundary 0 are called p-*cycles*, they form a subgroup $\mathsf{Z}_p(K)$ of $\mathsf{C}_p(K)$. The p-chains that are the boundary of $(p+1)$-chains are called p-boundaries and form a subgroup $\mathsf{B}_p(K)$ of $\mathsf{C}_p(K)$. The quotient group $\mathsf{H}_p(K) = \mathsf{C}_p(K)/\mathsf{B}_p(K)$ is called the p-th *homology* of K. Their ranks $\beta_p = \operatorname{rank} \mathsf{H}_p(K)$ of these abelian groups are called *Betti numbers*. The homologies and the Betti numbers are classical topological invariants of K.

In particular, a graph $G = (E, V)$ contains only 0-dimensional and 1-dimensional faces. It follows that its topological form is essentially described by the numbers β_0 and β_1 which are the only nonzero Betti numbers. Here β_0 is the number of connected components of G, and β_1 is the number of independent cycles of the graph (which is equal to $|E| - |V| + \beta_0$).

A *subcomplex* of K is a subset of simplices that is closed under the face relation. A *filtration* of K is a nested sequence of subcomplexes that starts with the empty complex and ends with the complete complex,

$$\emptyset \subset K_1 \subset K_2 \subset K_3 \subset \cdots \subset K_m = K.$$

In particular, to any weighted undirected graph $G = (V, E)$ and an increasing sequence $0 = t_0 \leq t_1 \leq \cdots \leq t_m$ such that t_m is greater or equal to the maximal edge weight in G, one can associate a filtration

$$\emptyset \subset G_{t_0} \subset \cdots \subset G_{t_m} = G, \tag{2}$$

where $G_{t_i} = (V, E_{t_i})$ and E_{t_i} consists of all edges of E with weight more or equal to t_i.

The p-th persistent homology of K is the pair of sets of vector spaces $\{\mathsf{H}_p(K_i) | 0 \leq i \leq l\}$ and maps $\{f_{i,j} : \mathsf{H}_p(K_i) \to \mathsf{H}_p(K_i) | 1 \leq i < j \leq l\}$, where the maps are induced by the inclusion maps $K_i \to K_j$.

Each persistent homology class α in this sequence is "born" at some K_i and "dies" at some K_j. One can visualize this as an interval $[i, j]$. The collection of all such intervals is called the *barcode* of the filtration. It is the most useful invariant of the filtration. Note that the information about the persistent homology classes is generally essential to calculate the barcode, whereas the information about the Betti numbers only is insufficient.

Still, in the case of the filtration associated to a weighted graph (2), the basis of H_0 (respectively, H_1) gives the intervals of the form $[0, t_i]$ (resp., $[t_i, t_m]$) only. Given a number $l = t_k$, the number of intervals of length at most l for H_0 (respectively, the number of intervals of of length at least $t_m - l$ for H_1) is therefore equal to the the Betti number $\beta_0(K_l)$ (resp., $\beta_1(K_l)$). We see that in this case the collection of the Betti numbers $\beta_i(K_{t_j})$ is sufficient to recover the barcode. Thus, we use just Betti numbers of the subgraphs K_{t_j} as the only topological invariants of our graphs.

Appendix B. Classifiers built by a single head

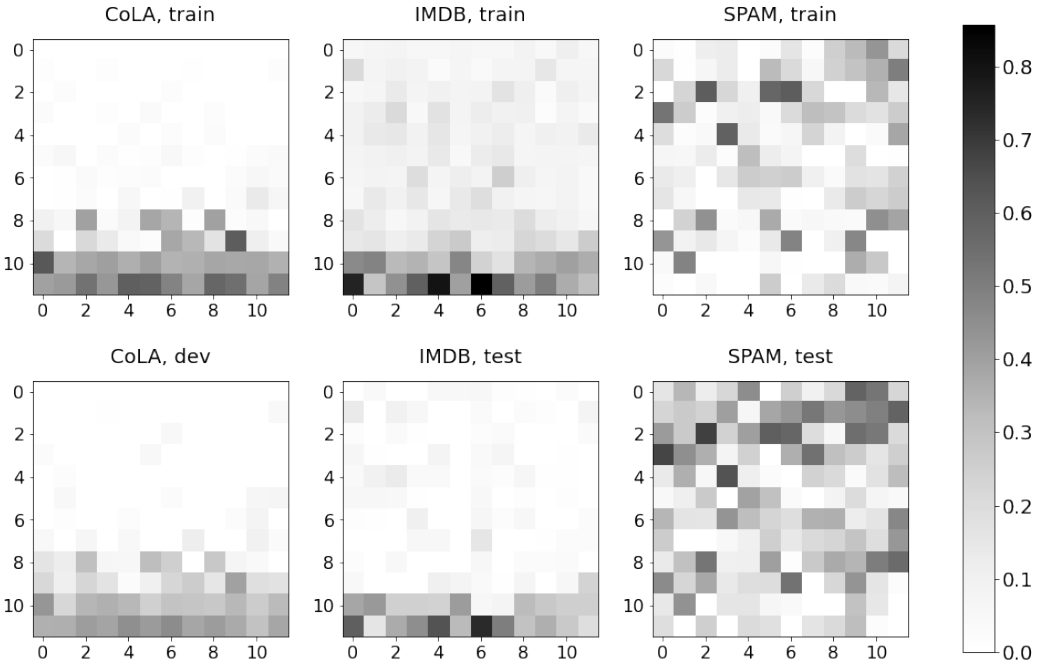

Figure 1: Matthew scores of predictions of linear classifiers, built upon particular attention heads. The number of the layer is displayed on the vertical axis. The number of the head inside the layer is displayed on the horizontal axis.

Figure 1 illustrates that the relevance of features, calculated on different heads, varies greatly from head to head on each task. It also shows that the same head can be more relevant for solving one task but less relevant for solving other ones. On the other hand, we can see similar patterns on the train and test/development sets for each task separately (in each column of Figure 1). This means that the head importance, derived from this score, is generalized to unseen examples and therefore can be used for feature selection.