Exploring Neural Language Models via Analysis of Local and Global Self-Attention Spaces

Blaž Škrlj
Jožef Stefan International
Postgraduate School
Jožef Stefan Institute, Slovenia
blaz.skrlj@ijs.si

Shane Sheehan
University of Edinburgh,
United Kingdom

Nika Eržen
Jožef Stefan Institute, Slovenia

Marko Robnik-Šikonja
University of Ljubljana, Slovenia

Saturnino Luz
University of Edinburgh,
United Kingdom

Senja Pollak
Jožef Stefan Institute, Slovenia

Abstract

Large pretrained language models using the transformer neural network architecture are becoming a dominant methodology for many natural language processing tasks, such as question answering, text classification, word sense disambiguation, text completion and machine translation. Commonly comprising hundreds of millions of parameters, these models offer state-of-the-art performance, but at the expense of interpretability. The attention mechanism is the main component of transformer networks. We present AttViz, a method for exploration of self-attention in transformer networks, which can help in explanation and debugging of the trained models by showing associations between text tokens in an input sequence. We show that existing deep learning pipelines can be explored with AttViz, which offers novel visualizations of the attention heads and their aggregations. We implemented the proposed methods in an online toolkit and an offline library. Using examples from news analysis, we demonstrate how AttViz can be used to inspect and potentially better understand what a model has learned.

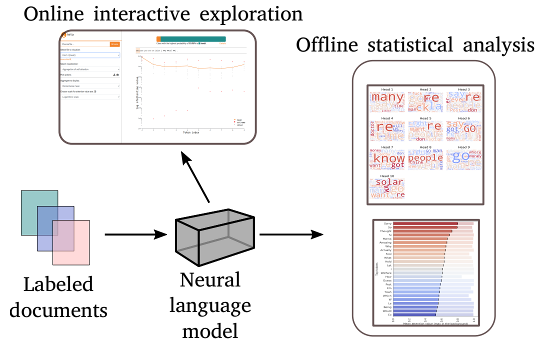


Figure 1: An overview of AttViz suite. The system consists of two main functional modules supporting online and offline visualizations. The online visualization (http://attviz.ijs.si; first part of the paper) offers direct exploration of token attention across the space of input documents; its purpose is anomaly detection and general inspection of the attention space (of trained models). The offline part of AttViz (second part of the paper) is a Python library that offers computationally more demanding statistical analyses, ranging from visualization of key tokens for each attention head, comparison of the attention head properties via FUJI integrals, and inspection of the attention distribution per-token basis.

1 Introduction

Currently the most successful machine learning approaches for text-related tasks predominantly use large *language models*. They are implemented with transformer neural network architecture (Vaswani et al., 2017), extensively pretrained on large text corpora to capture context-dependent meanings of individual tokens (Devlin et al., 2019; Liu et al., 2019; Yang et al., 2019). Even though training of such neural networks with hundreds of millions of parameters is long and expensive (Radford et al., 2019), many pre-trained models have been made freely available. This has created an opportunity to explore how, and why these models perform well on many tasks. One of the main problems with neural network models is their lack of *interpretability*. Even though the models learn the given task well, understanding the reasons behind the predictions, and assessing whether the model is

Proceedings of the EACL Hackashop on News Media Content Analysis and Automated Report Generation, pages 76–83
April 19, 2021 © Association for Computational Linguistics

susceptible to undue biases or spurious correlations is a non-trivial task.

Approaches to understanding black-box (non-interpretable) models include *post-hoc* perturbation methods, such as IME (Štrumbelj and Kononenko, 2010) and SHAP (Lundberg and Lee, 2017). These methods explain a given decision by assigning a credit to inputs (i.e. attributes or tokens) that contributed to it. These methods are not internal to the model itself and are not well adapted to the sequential nature of text-based inputs. Another way of extracting token relevance is the attention mechanism (Bahdanau et al., 2015; Luong et al., 2015) that learns token pair-value mappings, potentially encoding relations between token pairs. The attention of a token with respect to itself (called self-attention due its position on diagonal of the token attention matrix) offers certain insight into the importance of the token. Typically, a trained transformer network contains several attention heads, each bringing a different focus to the final decision of the network. Exploration of attention can be analytically and numerically cumbersome task, resulting in development of several approaches aimed at attention visualization collection.

As neural networks require numerical input, words are first transformed into a high dimensional numeric vector space, in a process called embedding that aims to preserve similarities and relations between words. Visualizations of embedding spaces is becoming ubiquitous in contemporary natural language processing. For example, Google's online Embedding Projector[1] offers numerous visualizations for technically non-savvy users, by projecting word vectors to low dimensional (human-understandable) spaces. While visualization of embedding spaces is already accessible, visualization of internal workings of complex transformer neural networks (e.g.,their self-attention mechanism) is a challenging task. The works of (Liu et al., 2018) and (Yanagimto et al., 2018) attempt to unveil the workings of black-box attention layers and offer an interface for human researches to learn and inspect their models. Liu et al. (2018) visualize the attention space by coloring it, and Yanagimto et al. (2018) visualize the self-attention with examples from a sentiment analysis.

In this work, we present AttViz, an online system that focuses exclusively on self-attention and introduces two novel ways of visualizing this property. The tool serves as an additional tool in the toolbox of a language model researcher, offering exploration of the learned models with minimal effort. AttViz can interactively aggregate the attention vectors and offers simultaneous exploration of the output probability space, as well as the attention space. A schematic overview of the proposed work is shown in Figure 1, and the main contributions are summarised as follows:

1. We present and describe AttViz, an interactive, online toolkit for visualization of the attention space of trained transformer neural language models.
2. We demonstrate the capabilities of AttViz on three problems: news classification, hate speech detection, and insults detection.
3. AttViz includes a stand-alone python library for offline analysis of the attention space, with the key focus on the relations between *the attention heads*.

The remainder of the paper is structured as follows. In Section 2, we discuss works related to the proposed AttViz approach. In Section 3, we present the key ideas and technical implementation of the online part of the AttViz system, including a use case on news classification. In Section 4, we discuss the capabilities of the AttViz library, available in an offline mode, and showcase its use on additional two datasets. In Section 5 we discuss capabilities and limitations of AttViz, present conclusions, and propose ideas for future work.

2 Background and related work on attention visualization

Neural language models are becoming the prevailing methodology for solving various text-related tasks, from entity recognition to classification. Visualization of the attention mechanism that is the key component of such models has recently emerged as an active research area due to an increased popularity of attention based methods in natural language processing. Recent deep neural network language models such as BERT (Devlin et al., 2019), XLNet (Yang et al., 2019), and RoBERTa (Liu et al., 2019) consist of multiple attention heads—separate weight spaces each associated with the input sequence in *a unique way*. These transformer language models consist of multiple attention matrices, all contributing to the final prediction. Visualising the attention weights

[1] https://projector.tensorflow.org/

from each of the attention matrices is an important component in understanding and interpreting these models.

The attention mechanism, which originated in the neural machine translation, lends itself naturally to visualisation. Bahdanau et al. (2015) used *heat maps* to display the attention weights between input and output text. This visualisation technique was first applied in machine translation but found its use in many other tasks. Rush et al. (2015) visualized an input sentence and the output abstractive summary, while Rocktäschel et al. (2016) showed an association between an input document and a textual entailment hypothesis on the output. In these heat map visualisations, a matrix is used to represent the token-token pairs and color intensity illustrates attention weights. This provides a summary of the attention patterns describing how they map the input to the output. For classification tasks, a similar visualisation approach can be used to display the attention weights between the classified document and the predicted label (Yang et al., 2016; Tsaptsinos, 2017). Here, the visualisation of attention often displays the input document with the attention weights superimposed onto individual words. The superimposed attention weights are represented similarly to heat map visualisations, using the color saturation to encode attention value. The neat-vision tool[2] encodes attention weights associated with input text in this manner. Similarly, the Text Attention Heatmap Visualization (TAHV[3]) which is included in the NCRF++ toolkit (Yang and Zhang, 2018) can be used to generate weighted sequences which are visualised using superimposed attention scores.

The purpose of the proposed AttViz is to unveil the attention layer space to human explorers in an intuitive manner. The tool emphasizes *self-attention*, that is, the diagonal of the token-token attention matrix which possibly corresponds to the *relevance* of individual tokens. Using different encoding techniques, attention weights across the layers and attention heads can be explored dynamically to investigate the interactions between the model and the input data. The AttViz tool differs from other tools in that if focuses on self-attention, thus allowing visualization of (attention-annotated) input token sequences to be carried out directly.

3 AttViz: An online toolkit for visualization of self-attention

AttViz is an online visualization tool that can visualize neural language models from the PyTorch-transformers library[4]—one of the most widely used resources for natural language modeling. The idea behind AttViz is that it is *simple to use* and *lightweight*, therefore it does not offer computationally expensive (online) neural model training, but facilitates the exploration of *trained* models. Along with AttViz, we provide a set of Python scripts that take as an input a trained neural language model and output a JSON file to be used by the AttViz visualisation tool. A common pipeline for using AttViz is outlined in Figure 1. First, a transformer-based trained neural network model is chosen to obtain predictions on a desired set of instances (documents or some other texts). The predictions are converted into the JSON format suitable for use with the AttViz tool, along with the attention space of the language model. The JSON file is loaded into the AttViz tool (on the user's machine, i.e. on the client side), where its visualization and exploration is possible. In Sections 3.1 and 3.3, we present the proposed self-attention visualizations, followed by an example of their use on the news classification task in Section 3.4.

3.1 Visualization of self-attention heads

We discuss the proposed visualization schemes that emphasize different aspects of self-attention. Following the first row that represents the input text, consequent rows correspond to attention values that represent the importance of a given token with respect to a given attention head. As discussed in the empirical part of the paper (Section 3.4), the rationale for this display is that typically only a certain number of attention heads are activated (colored fields). Thus, the visualization has to entail both the whole attention space, as well as emphasize individual heads (and tokens). The initial AttViz view offers sequence-level visualization, where each (byte-pair encoded) token is equipped with a self-attention value based on a given attention head (see Figure 4; central text space). The same document can also be viewed in the "aggregation" mode (Figure 2), where the attention sequence is shown across the token space. The user can interactively explore how the self-attention varies for individ-

[2] https://github.com/cbaziotis/neat-vision

[3] https://github.com/jiesutd/Text-Attention-Heatmap-Visualization

[4] https://github.com/huggingface/transformers

ual input tokens, by changing the scale, as well as the type of the aggregation. The visualization can emphasize various aspects of the self-attention space.

The third proposed visualization (Figure 3) is the overall distribution of attention values across the whole token space. For each consequent token, the attention values are plotted separately, resembling a time series. This visualization offers an insight into *self-attention peaks*, i.e. parts of the attention space around certain tokens that potentially impact the performance and decision making process of a given neural network. This view can emphasize different aggregations of the attention vector space for a single token (e.g., mean, entropy, and maximum). The visualization, apart from the mean self-attention (per token), offers the information on maximum and minimum attention values (red dots), as well as the remainder of the self-attention values (gray dots). In this way, a user can explore both the self-attention peaks, as well as the overall spread.

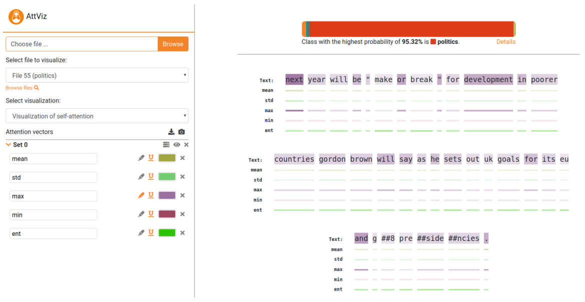
3.2 Comparison with state-of-the-art

In the following section, we discuss similarities and differences between AttViz and other state-of-the-art visualization approaches. Comparisons are summarized in Table 1.

Novel functionality introduced by AttViz include the capability to aggregate the attention vectors with four different aggregation schemes, offering insights both into the average attention but also its dispersion around a given token. The neat-vision project[5] is the closest to AttViz in terms of functionality. However, a few differences should be noted. First, neat-vision is not directly bound to the PyTorch transformers library, requiring additional pre-processing on the user-side. Second, switching between the sequence and aggregate view is faster and more emphasized in AttViz, as it offers a more general overview of the attention space.

3.3 Aggregation of self-attention

The self-attention is captured in the matrix $A \in \mathbb{R}^{h \times t}$, where h is the number of attention vectors and t the number of tokens. Aggregation operators are applied the second dimension of the attention matrix A (index j). We denote with P_{ij} the probability of observing A_{ij} in the j-th column. The m_j corresponds to the number of unique values in that column. The proposed schemes are summarized in Table 2. The attention aggregates are visualized as part of the the aggregate view (see Figure 4). For example, the mean attention is plotted as a line along with the attention space for each token, depicting the *dispersion* around certain parts of the input text.


AttViz

Choose file ... Browse

Select file to visualize: File 55 (politics)

Select visualization: Visualization of self-attention

Attention vectors

Set 0

mean

std

max

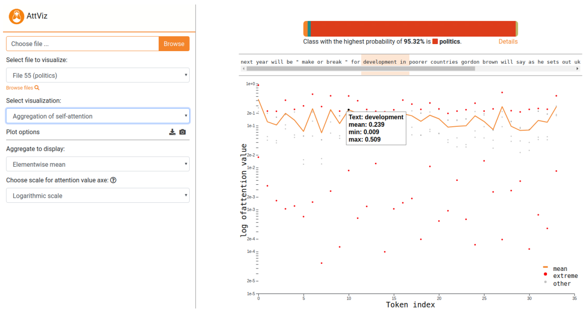
min

ent

Class with the highest probability of **95.32%** is 🟥 **politics**. Details

Browse files
Text: next year will be " make or break " for development in poorer
mean
std
max
min
ent
Text: countries gordon brown will say as he sets out uk goals for its eu
mean
std
max
min
ent
Text: and g ##8 pre ##side ##ncies .
mean
std
max
min
ent

Figure 2: Visualization of aggregations. The document was classified as a politics-related topic; the aggregations emphasize tokens such as “development”,“uk” and “poorer”. The user can highlight desired head information – in this example the maximum attention (purple) is highlighted.


AttViz

Choose file ... Browse

Select file to visualize: File 55 (politics)

Select visualization: Aggregation of self-attention

Plot options

Aggregate to display: Elementwise mean

Choose scale for attention value axe: Logarithmic scale

Class with the highest probability of **95.32%** is 🟥 **politics**. Details

next year will be " make or break " for development in poorer countries gordon brown will say as he sets out uk

Text: development
mean: 0.239
min: 0.009
max: 0.509

Figure 3: The interactive series view. The user can, by hoovering over the desired part of the sequence, inspect the attention values and their aggregations. The text above the visualization is highlighted automatically.

3.4 Example: News visualization

In this section, we present a step-by-step use of the AttViz system along with potential insights a user can obtain.

The examples are based on the BBC news data set[6] (Greene and Cunningham, 2006) that contains 2,225 news articles on five different topics (business, entertainment, politics, sport, tech). The documents from the dataset were split into short segments. The splits allow easier training (manage-

[5] Available at https://github.com/cbaziotis/neat-vision

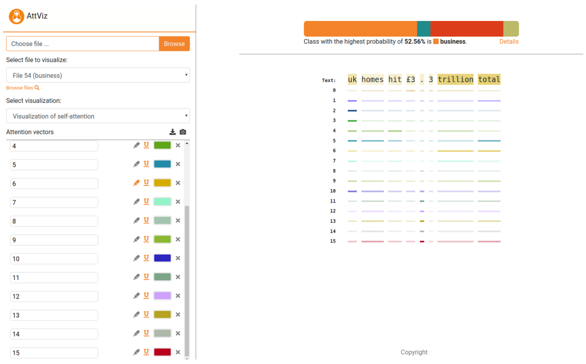
[6] https://github.com/suraj-deshmukh/BBC-Dataset-News-Classification/blob/master/dataset/dataset.csv

Approach	AttViz (this work)	BertViz (Vig, 2019)	neat-vision	NCRF++ (Yang and Zhang, 2018)
Visualization types	sequence, aggregates	head, model, neuron	sequence	sequence
Open source	✓	✓	✓	✓
Language	Python + Node.js	Python	Python + Node.js	Python
Accessibility	Online	Jupyter notebooks	Online	script-based
Sequence view	✓	✓	✓	✓
Interactive	✓	✓	✓	✗
Aggregated view	✓	✗	✗	✗
Target probabilities	✓	✗	✓	✗
Compatible with PyTorch Transformers? (Wolf et al., 2020)	✓	✓	✗	✗
token-to-token attention	✗	✓	✗	✓

Table 1: Comparison of different aspects of the attention visualization approaches.

Table 2: Aggregation schemes used in AttViz. The A represents a real valued (attention) matrix.

Aggregate name	Definition
Mean(j) (mean)	$\frac{1}{h}\sum_i A_{ij}$
Entropy(j) (ent)	$-\frac{1}{m_j}\sum_{i=0}^{h} A_{ij}\log A_{ij}$
Standard deviation(j) (std)	$\sqrt{\frac{1}{h-1}\sum_i (A_{ij}-\overline{A_{ij}})^2}$
Elementwise Max(j) (max)	$\max_i(A_{ij})$
Elementwise Min(j) (min)	$\min_i(A_{ij})$


Choose file ...

Select file to visualize: File 54 (business)

Select visualization: Visualization of self-attention

Class with the highest probability of 52.56% is business.

Text: uk homes hit £3 . 3 trillion total

Figure 4: Visualization of all attention heads. The sixth heads's self attention is used to highlight the text. The document was classified as a business-related, which can be linked to high self attention at the "trillion" and "uk" tokens. Compared to the first two examples (Figures 2 and 3), the network is less *certain* – in this example, the business (orange) and politics (red) classes were predicted with similar probabilities (orange and red parts of the bar above visualized text).

able sequence lengths), as well as easier inspection of the models. We split the dataset into 60% of the documents that were used to fine-tune the BERT-base (Devlin et al., 2019) model, 20% for validation and 20% for testing. The Nvidia Tesla V100 GPU processor was used for these experiments. The resulting model classified the whole documents into five categories with 96% accuracy, which is comparable with the state-of-the-art performance (Trieu et al., 2017). For prediction and visualisation, we used only short segments. The fine-tuning of the BERT model follows examples given in the PyTorch-Transformers library (Wolf et al., 2020). The best-performing hyper parameter combination used 3 epochs with the sequence length of 512 (other hyper parameters were left at their default values). While we have used BERT, similar explorations could be made for more recent larger models such as XLNet (Yang et al., 2019) that might could produce better classification accuracy.

The user interface of AttViz is displayed in Figures 2, 3, and 4. In the first example (Figure 3), the user can observe the main view that consists of two parts. The leftmost part shows (by id) individual self-attention vectors, along with visualization, aggregation and file selection options. The file selection indexes all examples contained in the input (JSON) file. Attention vectors can be colored with custom colors, as shown in the central (token-value view). The user can observe that, for example, the violet attention head (no. 5) is active, and emphasizes tokens such as "development", which indicates a politics-related topic (as correctly classified). Here, the token (byte-pair encoded) space is shown along with self-attention values for each token. The attention vectors are shown below the token space and aligned for direct inspection (and correspondence).

In Figure 4, the user can observe the same text segment as an attention series spanning the input token space. Again, note that tokens, such as "trillion" and "uk" correspond to high values in a subset of the attention heads, indicating their potential importance for the obtained classification. However, we observed that only a few attention heads activate with respect to individual tokens, indicating that other attention heads are not focusing on the tokens themselves, but possibly on *relations* between them. This is possible, and the attention matrices contain such information (Vig, 2019). However, as mentioned earlier, the study of token relations is not the focus of this work. As self-attention information can be mapped across token sequences, emphasizing tokens that are of relevance to the classification task at hand, we see AttViz as being the most useful when exploring models used for text classification tasks, such as hate speech detection and sentiment analysis, where individual tokens contain the key information for classification.

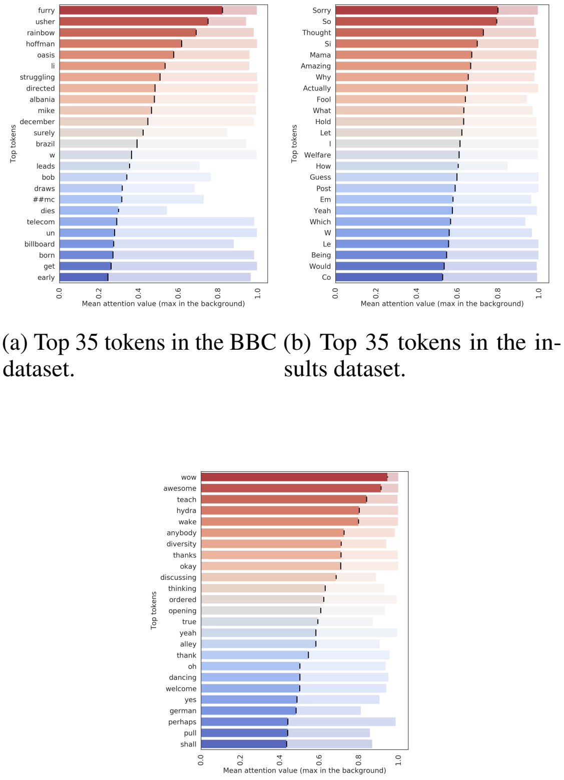
The example above shows how different attention heads detect different aspects of the sentence, even at the single token (self-attention) level. The user can observe that the next most probable category for this topic was politics (red color), which is indeed a more sensible classification than, for instance, sports. The example shows how interpretation of the attention can be coupled with the model's output for increased interpretability.

4 AttViz library: statistical analysis of the attention space

In Section 3 we presented how the online version of AttViz can be used for *direct analysis* of model output (in the JSON format). Albeit suitable for quick inspections, the online system has its limitations such as poor support for computationally more intensive types of analysis (in terms of waiting times), and the lack of customized visualization tools accessible in the Python ecosystem. To address these aspects, we developed AttViz library that offers more detailed analysis of a given neural language model's properties. The library operates on the same JSON structures as the online version and is compatible with the initial user input. We demonstrate the analytical capabilities of our visualization tools on three datasets. The BBC news classification was already presented in Section 3.4.

4.1 Dissecting the token space

The first offline functionality is a barplot visualization that offers insight into relevant aspects of the attention distribution at token level. Whilst understanding the attention peaks is relevant for direct inspections (Section 3), the attention space of a given token can be contextualized on the dataset level as well. The AttViz library offers fast visualization of the mean and spread of attention distributions, simultaneously showing the attention peaks for individual tokens. We visualized the distribution for three classification datasets (Figure 5): BBC news (5a), insults[7] (5b), and hate speech comments (5c)[8].



(a) Top 35 tokens in the BBC dataset.

(b) Top 35 tokens in the insults dataset.

(c) Top 35 tokens in the hate speech dataset.

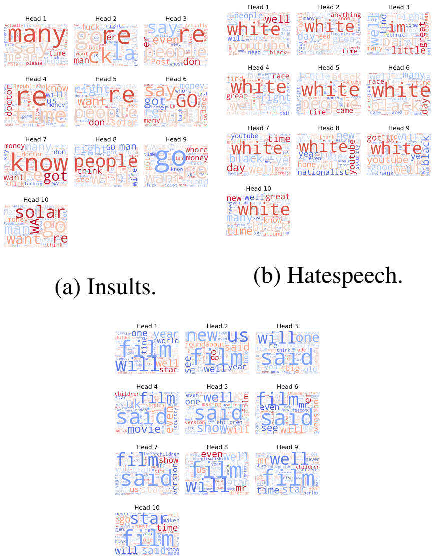
Figure 5: Visualization of the 35 most attended-to tokens for the three inspected data sets. Interestingly, the attention peaks of tokens (maximum, in the background) all take high values, albeit lower-ranked tokens are on average characterized by lower mean attention values.

The proposed visualizations present top k tokens according to their mean attention throughout the whole dataset. It is interesting to observe, that the insults and hate speech data sets are not completely characterized by swear words or similar single-token-like features. This potentially indicates that the attention tries to detect interactions between the byte-pair encoded tokens, even for data sets where the attention could be focused on single tokens. It is interesting to observe that the terms with the highest attention are not necessarily keywords or other tokens carrying large semantic meaning. Similarly, the high maxima indicate that the emphasis of the tokens is very contextual, and potentially not as informative for global aggregation.

[7] https://www.kaggle.com/c/detecting-insults-in-social-commentary/overview

[8] https://github.com/aitor-garcia-p/hate-speech-dataset

4.2 Visualization of attention head focus



(a) Insults.

(b) Hatespeech.

(c) BBC news.

Figure 6: The distribution of tokens over individual attention heads for the three datasets summarised with word clouds.

Contemporary neural language model architectures comprise multiple attention heads. These separate weight spaces capture distinct aspects of the considered learning task. Even though the weight spaces are easily accessible, it is not trivial to convert the large amount of information into a quick-to-inspect visualization. With the proposed visualization, shown in Figure 6, we leverage word clouds (Kaser and Lemire, 2007) to reveal human-understandable patterns captured by separate attention heads and display this information in a compact way.

5 Discussion and conclusions

As AttViz is an online and offline toolkit for attention exploration, we discuss possible concerns regarding its use, namely: privacy, memory and performance overheads, and coverage. Privacy is a potential concern for most web-based systems. As currently AttViz does not employ any anonymization strategy, private processing of the input data is not guaranteed. While we intend to address this issue in furture work, a private installation of the tool can be done to get around this current limitation. AttViz uses the users' computing capabilities, which means that large data sets may cause memory overheads when a large number of instances is loaded (typically several million). Such situations are difficult to address with AttViz and similar web-based tool, but users can filter instances before using them in AttViz and explore a subset of the data (e.g., only (in)correctly predicted instances, or certain time slot of instances). Finally, AttViz is focused on the exploration of *self-attention*. This is not the only important aspect of a transformer neural network, but it is the one, where visualisation techniques have not yet been sufficiently explored. Similarly to the work of (Liu et al., 2018), we plan to further explore potentially interesting *relations* emerging from the attention matrices.

6 Availability

The software is available at `https://github.com/SkBlaz/attviz`.

Acknowledgements

We acknowledge European Union's Horizon 2020 research and innovation programme under grant agreement No 825153, project EMBEDDIA (Cross-Lingual Embeddings). The first author was also funded by Slovenian Research Agency as a young researcher.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Derek Greene and Padraig Cunningham. 2006. Practical solutions to the problem of diagonal dominance in kernel document clustering. In *Machine Learning, Proceedings of the Twenty-Third International Conference (ICML 2006), Pittsburgh, Pennsylvania, USA, June 25-29, 2006*, volume 148 of *ACM International Conference Proceeding Series*, pages 377–384. ACM.

Owen Kaser and Daniel Lemire. 2007. Tag-cloud drawing: Algorithms for cloud visualization. In *Procedings of WWW Workshop on Tagging and Metadata for Social Information Organization*.

Shusen Liu, Tao Li, Zhimin Li, Vivek Srikumar, Valerio Pascucci, and Peer-Timo Bremer. 2018. Visual interrogation of attention-based models for natural language inference and machine comprehension. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 36–41, Brussels, Belgium. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Scott M. Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 4765–4774.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8).

Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tomás Kociský, and Phil Blunsom. 2016. Reasoning about entailment with neural attention. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389, Lisbon, Portugal. Association for Computational Linguistics.

Lap Q. Trieu, Huy Q. Tran, and Minh-Triet Tran. 2017. News classification from social media using twitter-based doc2vec model and automatic query expansion. In *Proceedings of the Eighth International Symposium on Information and Communication Technology*, SoICT 2017, pages 460–467.

Alexandros Tsaptsinos. 2017. Lyrics-based music genre classification using a hierarchical attention network. *CoRR*, abs/1707.04678.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Jesse Vig. 2019. Visualizing attention in transformer-based language representation models. *CoRR*, abs/1904.02679.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

H. Yanagimto, K. Hashimoto, and M. Okada. 2018. Attention visualization of gated convolutional neural networks with self attention in sentiment analysis. In *2018 International Conference on Machine Learning and Data Engineering (iCMLDE)*, pages 77–82.

Jie Yang and Yue Zhang. 2018. NCRF++: An open-source neural sequence labeling toolkit. In *Proceedings of ACL 2018, System Demonstrations*, pages 74–79, Melbourne, Australia. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5754–5764.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489, San Diego, California. Association for Computational Linguistics.

Erik Štrumbelj and Igor Kononenko. 2010. An efficient explanation of individual classifications using game theory. *Journal of Machine Learning Research*, 11:1–18.