REDACBERT: Redaction through Ensembled Domain-Aware Context via BERT

Rudy Venguswamy , Chelsea Shu and Jennifer Patterson

Abstract. In this paper we propose a novel pipeline to redact information related to any given topic from a document. We implement an iterative word redaction scheme by ensembling BERT QA with BERT topic embedding cosine similarity averaging, generating fruitful comparisons on candidate token embeddings which allows for an effective decision function. Our methods allow for the user to combine their domain knowledge of a topic into the more generalized automated redaction pipeline. The model is domain and corpus agnostic, but in this paper we demonstrate and evaluate our model in context of deracializing a corpus of legal documents. Our technique achieves 100% precision compared to a baseline standard search-and-redact approach, which achieved 63.15% precision on our (arguably small) dataset.

1. Background

While our pipeline functions for eliminating information related to any topic, we chose to develop it on a corpora of legal cases, and focused our redaction efforts on ethnic and racial information. We were inspired to do this because there are practical motivations for having a neutralized (‘debiased’, ‘objective’) body of case law. One practical, albeit idealistic, motivation is to use this neutralized dataset when researching case law – redacting race indicators may allow research attorneys to focus on strictly the legal papertrail, without race or ethnicity playing a role in analysis.

A second interesting use case would involve creating one case outcome prediction model trained on a neutralized case set, and one case outcome prediction model trained on an original case set. The inverse could also be viable; a case outcome prediction model can be trained on other (untouched) data, and the neutralized set and original set can be run through and their outcomes predicted; in this case, a discrepancy in predicted outcomes on the neutralized versus original dataset may point us to cases where racial biases or prejudices may have influenced court decisions.

2. Prior Work

Karve et al. examines the use of the debiasing conceptor, a “mathematical representation of subspaces that can be operated on and composed by logic-based manipulations”, to debias word embeddings (Karve et al, 2019). The authors find that their approach can successfully remove racial biases, and that their debiasing conceptors work well with BERT due to the large number of embeddings. However, the debiasing conceptor occasionally generates bias in BERT. It should be noted this paper uses the Word Embedding Association Test (WEAT) to measure bias of names and considers debiasing in the context of ethnic names. Its primary focus does not involve debiasing outside the scope of ethnic names while our paper’s focus is on races itself.

Manzini et al. introduces the idea of a bias subspace formed from the theoretical motivation that multiple social classes based on components of word embeddings are linearly separable (Garg et

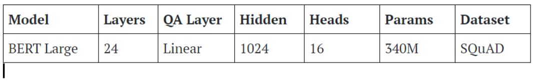
2451-8484 © 2020 – IOS Press and the authors. All rights reserved

al., 2018). They theorize that a single component of embeddings exists that can capture multiclass bias by framing multiclass bias as a one-versus-rest classifier. Furthermore, Manzini introduces the term Mean Average Cosine similarity (MAC) that provides the theoretical backing for our final redaction decision function separator. We draw on the Manzini paper for inspiration and extend the notion of linearly separable multiclass bias detection with Word2Vec embeddings to BERT. Furthermore, the authors note their solution depends upon a ground truth dataset of unbiased names and vocabulary, which we attempt to mitigate, though not allege to solve entirely, by using BERT embeddings itself.

3. Methodology

3.1. General Pipeline

The pipeline that we used to debias documents relies on a BERT Question Answering model (BERT QA) to infer the tokens that relay racial or ethnic information. The uncased large BERT QA model was pre-trained with whole word masking tokenizers on the SQuAD dataset as per the defaults prescribed in the Huggingface library run_squad function baseline.



Model	Layers	QA Layer	Hidden	Heads	Params	Dataset
BERT Large	24	Linear	1024	16	340M	SQuAD

Without access to an expansive legal Q&A document-based dataset, attempts to use transfer learning by fine-tuning a base BERT model on legal data and appending the linear QA layer did not succeed because of inconsistencies within the linear layer weights. We theorize that the failure occurs due to changes in the penultimate layer of the BERT model during SQuAD fine-tuning. As such, we opted to continue to use the BERT model fine-tuned on SQuAD.

To begin the redaction process, the model is asked a pointed question (we use 'What is the race of the person?' and 'What is the race of the defendant?') about a case text. BERT QA chooses the best answer and funnels it to our embedding space decision function, the output of which determines whether or not the chosen answer is actually redacted. If the answer is redacted by the embedding space decision function, the text and question are run through BERT QA again. This process happens iteratively until BERT QA is only returning answers which do not trigger a redaction response in the decision function.

3.2. The "512 Problem"

The pre-trained BERT model restricts its input size to 512 tokens; however, the average size of the case law documents are more than 10-15 times that size. We explored abstractive summarization as a solution to this. Our expectation was that abstractive summarization would omit superfluous information from the legal document and reduce input dimensionality, acting as a roundabout form of principal component analysis that would maintain most relevant context. We also thought that the abstractive model would resolve complex statements that imply race and ethnicity into simpler, easy-to-detect statements for BERT QA. However, the abstractive summarization model was not able to resolve these complex racial cues, nor significantly reduce the size of case texts without excluding valuable context, so we abandoned it.

The next approach was to use XLNet due to its ability to process high-volume inputs; however, XLNet does not have a pretrained model on the SQuAD dataset, so in the interest of time we opted to continue using BERT.

Our final approach to solving the "512 problem" was to break apart and then iteratively scan through the document. During processing, each document is broken up into chunks of 462 tokens to allow a 50 token buffer for the question

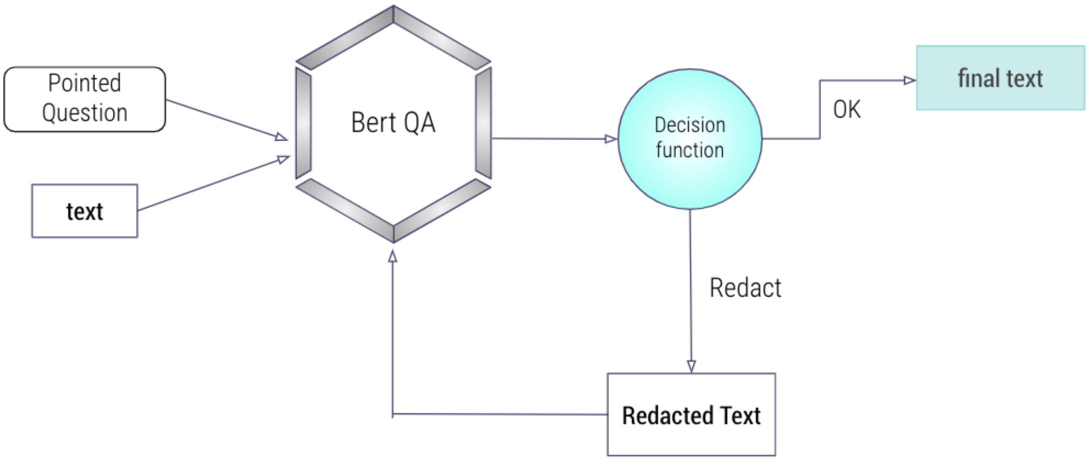


Fig. 1. A general pipeline for text debiasing that relies on BERT QA

portion of each query. Furthermore, we created a 50 token overlap between chunks to account for answers appearing at the end of a chunk. Each chunk is evaluated for the answer by BERT QA, and the answer with the highest confidence is chosen as the final answer for the whole document (see Fig 2).

When we adapted the standalone QA model into our debiasing pipeline, we realized that we could redact problematic words in-place as the chunks were iterated through, since we are looking for all cases of inferred racial information instead of just one answer. This precludes the need for answer comparison between chunks.

3.3. Debiasing

We attempted a sequence of four increasingly effective approaches to implement our debiasing pipeline: (1) a dictionary approach, (2) spaCy dependency parsing approach, (3) naive direct QA auto-redaction, (4) Ensembling Direct QA auto-redaction with BERT Attention Mechanism Heuristic, and finally, (5) a similar naive direct QA auto-redaction with added target-candidate embedding comparison.

3.3.1. Previous Attempts

Dictionary Approach. In this approach, we compared the answers to questions such as "What is the race of the defendant?" to a dictionary of words related to race or ethnicity. If the answer matched a word in the dictionary of "problem" words, we redacted them. This method did not work well because it would remove all instances of a word if there was even only one problematic instance of it, regardless of whether that word was related to an object or a person. For example, in the sentence "Andrew Johnson, a white man, was found in a white Honda Civic," the dictionary approach would redact both instances of white, rather than redacting only the first instance (the "white walls" problem). This dictionary approach worked only a little better than a "search-and-replace" approach.

SpaCy Dependency Parsing Approach. The next approach involved using SpaCy's dependency parsing capabilities to determine when a

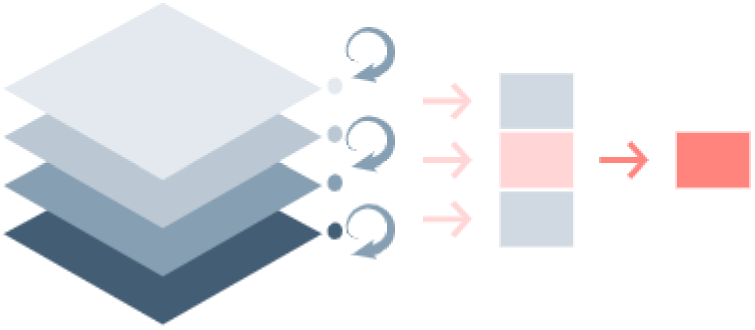


Fig. 2. An illustration of our 512 scanning. Each chunk is evaluated for an answer, and the answers are then compared. In the above abstract example, the answer represented by the pink box has the highest confidence metric and is selected to be the answer for the whole document.

potential problem word, as inferred by BERT QA, was related to a person entity. This approach allows an analysis of grammatical structures and identification of relationships between words, and in early forms worked much better than our simple dictionary approach for tackling the "white Honda Civic" problem. However, using it perfectly required a significant amount of complicated tree-climbing, i.e., navigating complex parent-child relationships in spaCy's dependency structure to determine a word's relation to person-entities.

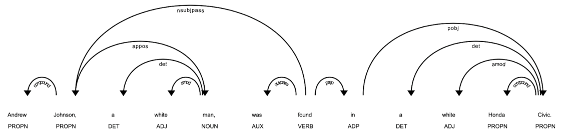


Fig. 3. A depiction of spaCy dependency parsing. This illustration demonstrates the parent-child grammatical relationships between words.

Naive Direct QA Auto-redaction. Our next approach involved directly redacting the answer every time the QA model is called. Fortunately for us, this approach generally avoided the "white Honda Civic" problem and avoided tree-climbing complications; this was partially because the redaction was done within our model call, so the problem answer was still represented, and therefore removed, as a unique token at the time of redaction. However, this approach had no limitations on redaction once an answer was selected–if the QA model continued to find answers after the intended problem text was redacted, those other answers were also redacted, resulting in unnecessary information loss.

Ensembling Direct QA Auto-redaction with BERT Attention Mechanism Heuristic. We considered adding a check prior to an answer being redacted via BERT QA. We investigated the attention weights to see if BERT attends to named entities or sentence structure, similar to spaCy. However, we could not discern which heads, if any, were responsible for structure. (Appendix, Fig. 5 & Fig. 6)

3.3.2. Current Methodology

REDACBERT - Ensembling Direct QA Auto-redaction with Novel BERT Embedding Comparison. Our final approach builds upon the former by using cosine similarity to compare the embeddings of potential answers to baseline term embeddings that we felt functioned as a representation of the concept of race and ethnicity (e.g. 'race', 'ethnic', 'Hispanic').

If a candidate for redaction meets a certain average similarity threshold compared to our baseline terms, the candidate will then be redacted. The baseline terms we used come from a list of the top 100 represented races and ethnicities in the United States, according to Wikipedia and the U.S. Census. This approach proved to be the most successful because it was more conservative and therefore less aggressive when choosing words to redact, resulting in reduced information loss.

With our SQuAD fine-tuned model we conduct iterative forward passes in our model to generate candidates for redaction:

Let function F be defined as the composite function of the$\beta \circ T$ where βrepresents the linear transformation of the embeddings E generated from Transformer Model $T : dimE < R^{1024}$ from word tokens. The multiple tokens per word explains the need to average across multiple candidate tokens. Our iterative process is outlined here:

1. $C = Fn(Q, D_n) : D_n \subseteq D_{n-1}$
2. $C_E = \sum_{j=0}^{C} \sum_{j=0}^{S} \frac{\frac{l_i * c_j}{||l_i|| * ||c_j||}}{||C|| * ||S||} > \lambda$
3. $D_{n+1} = D_n : C_E \not\subset D_n$

where C represents the candidate words to be pruned; $C_E$represents the candidates from our embedding space decision function; Q represents our question on race; D represents the set of words in a document that are to be iteratively searched for candidates to be redacted; S represents the set of words provided by the user to capture a target domain for redaction; and λ represents the aggressiveness hyperparameter, which determines how thorough the redaction should be. The number of iterations n is unlimited and only stops once C_E = $\varnothing$or once C= $\varnothing$, in which case $F(Q, D_n)$= [REDACTED]. Note that $C_E \subseteq C$ as a result of the operation.

As outlined in Equation 2, we chose to forego the latent space representation of the notion of race in favor of an average cosine similarity, averaging similarity scores across multiple tokens. We considered conducting a Principal Component Analysis (PCA) of our set of ethnicity/race-related words, similar to that of previous authors who removed the first principal component of a set of racialized names to debiase them (Wang et al., 2019). However, it became clear that the dimensionality reduction suffered as more words were added to the list, as reducing the dimensionality introduced a further hyperparameter for the size of the latent space and resulted in only a single comparison per candidate against this space. Furthermore, we have access to something very roughly approximating the ground truth of what PCA on a complete set of racial and ethnic terms would yield- an embedding representing the ethnoracial connotation of "race." As a result, we implement a trick to incorporate this central embedding by appending it to our topic list itself, creating a centroid, albeit one that captures race with its many different connotations. We also add the word "ethnicity" which is more unambiguous to proxy this groundtruth centroid coordinate. By averaging cosine similarity across candidate tokens and our topic list, we emulate the implementation of the MAC score described in Manzini et. al. This opens our model to be sensitive to matches with certain subsets of races, but allow for multiple fuzzy match comparisons, including those with the centroids, that contribute to the similarity score of the candidate token. The use of the token averages is motivated by numerous previous papers (Karve et. al 2019). Unlike previous authors who form their embeddings by summing the last four layers, however, we form our embeddings by averaging from the second to penultimate layer, inclusive, of the BERT QA model. This embedding representation was found to be the most effective empirically but has a history of discussion (Ma et. al, 2019).

4. Evaluation

In order to evaluate the effectiveness of our pipeline, we tested it on 16 relevant excerpts from a sample of 43 cases from Harvard Law School's Caselaw Access Project. All of the cases were from the state of California, and some of them contained racial information while others did not. For this sample, we manually generated an ideal debiase key, i.e. what we would want our pipeline to be able to output in the most ideal case. We then tested our debiase key against both our minimal baseline (a basic search-and-replace with a set of the top 100 races and ethnicities in the US, according to the US Census Bureau) and our fifth and best pipeline, the direct auto-redaction with candidate-target embedding comparison.

We scored the two methods on their true and false positives and negatives, translated these raw scores into precision and recall scores, calculated their F1-score and plotted their respective AUC-ROC curves in the Results section below. We focus on precision and recall scores rather than accuracy due to the significant class imbalance between redacted and unaffected words inherent to a redaction problem. Accuracy metrics are skewed by this imbalance.

We would like to note that originally, we had a larger evaluation set that included the entire text body for 50 cases. After annotating the set with the desired redactions, there were technical issues running it against the model outputs, likely probably due to white space formatting in the text files we used. For this reason and due to time constraints, we had to select just a few excerpts from the cases to recode and run through the model. Unfortunately, this decreased the volume and variability of "problems" we would expect the pipeline to run into in real-life application. Also for this reason, there happened to be no cases in which it was possible for our search-and-redact baseline to produce a false negative resulting in a perfect recall score, despite the fact we would expect such cases to occur in real-life application.

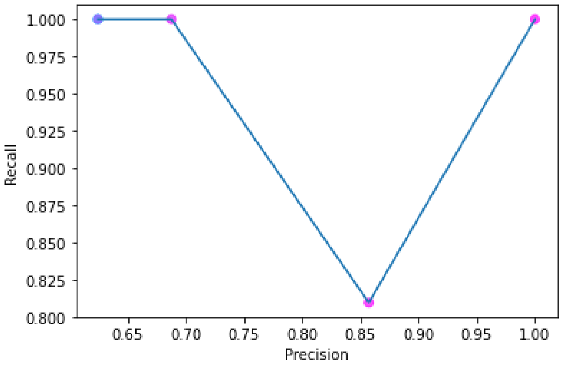


Fig. 4. AUC-ROC Curve

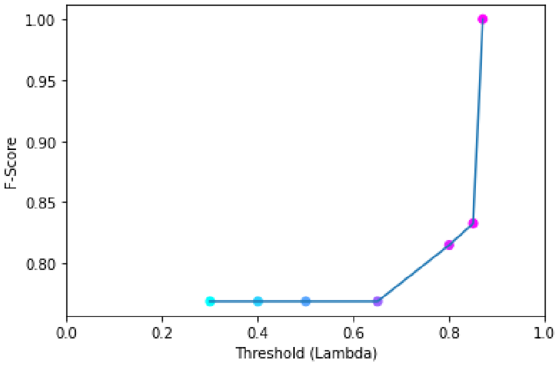


Fig. 5. Threshold (λ) vs. F1

5. Results

Figure 4 shows the relationship between precision and recall on our evaluation data. We tested 9 thresholds as seen in the table above. For threshold values below 0.65, the BERT QA model "picked up the slack" in that even though the embedding threshold lambda was lenient (allowed for lower similarity), the QA model's candidate suggestions were good enough to still receive a precision score of 0.62. We expected the recall to drop as precision increased. Recall drops at precision 0.85, but spikes back up at 0.87. We theorize this abnormal AUC-ROC curve shape is due to the complex nature of the ensemble set up, along with the complexity that emerges from iterative word pruning through our ensemble network. We find that at λ=0.87 our model performs

Table 1

Results Table

Threshold (Lambda)	Precision	Recall	F1
Baseline Search & Replace	**0.6315**	1	0.7741
0.3	0.6243	1	0.7687
0.4	0.6243	1	0.7687
0.5	0.6243	1	0.7687
0.65	0.6243	1	0.7687
0.8	0.6875	1	0.8148
0.85	0.8571	0.8095	0.8326
0.87	1	1	1
0.9	nan	nan	nan
0.95	nan	nan	nan

the best: on our limited evaluation set, it achieves perfect precision and recall. This value, while promising, should be interpreted cautiously due to the small size of our evaluation dataset. Any higher values for lambda result in extreme values (NaN).

In Figure 5, we plot the threshold lambda against the F-Measure for our evaluation set. We see a positive increasing relationship between and the F-Measure. This suggests that at λ=0.87 we achieve a reasonable balance between the embedding throttling BERT QA but not doing so to the point no values pass its criteria.

6. Conclusion

In this paper we have discussed the architecture and application of our word redaction pipeline, used in the legal case debiasing context. While our pipeline saw great success relative to a search-and-redact baseline, it is far from perfect (especially, presumably, on larger and more complex evaluation sets), and there are several possible improvements that will be discussed below.

6.1. Improvements

6.1.1. Conducting information loss analysis on our outputs

In order for our resulting dataset to be useful for precedent analysis or outcome modeling, it is imperative that vital case details and context are not removed. Therefore, we propose performing an information-context analysis on case outputs before and after redaction to estimate information loss; this score would then be compared to that of the search-and-redact approach.

6.1.2. Account for racially indicative names and locations

While our current model currently redacts information related to race or ethnicity, it does not account for racially-indicative names or locations. We would like to adapt our model to censor and replace a racially-indicative name with a neutralized form (e.g. “the subject,” “the defendant”). Of course, redacting only “ethnic” names poses a problem, since any redacted name can be then assumed to be a minority. Therefore, all names must be redacted and replaced with a unique placeholder identifier consistent throughout the document.

6.1.3. Pretraining pre-QA BERT on our legal data

We noticed that our model fails to recognize some legal terms, which makes sense due to the fact that it was pre-trained on the Stanford Question Answering Dataset (SQuAD) rather than on legal data. This failure to recognize some common legal terms is problematic because there are cases when a race-related or ethnicity-related word is dependent on a person-entity represented by a legal term (e.g. “jurors”, “witnesses”). Without being able to recognize this dependency, our debiasing pipeline might fail to redact sensitive racial information. A potential solution would be to pre-train our model on legal corpora after training it on SQuAD to allow for the model to learn and incorporate legal terms into its vocabulary.

6.2. Other Considerations

As a disclaimer, we would like to acknowledge that this debiasing method should likely not be used for all case types. For example, cases concerning protected classes or racially-motivated hate crimes may require that racial information be included in order to have any real use in analysis.

It is also important to consider in the outputs of our current pipeline, a redaction tag may indicate to a reader or model that an individual is part of a minority race. This is because on average, race is mentioned more often in cases that involve a member of a minority race than in cases that involve white individuals. A potential solution to this is omitting the in-place [REDACTED] tag entirely, although we kept it in order to facilitate our evaluation scoring.

7. Acknowledgements

We would like to acknowledge Hugging Face and Harvard Law School's Caselaw Access Project.

8. References

Devlin, Chang, Lee & Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.

Karve, Ungar, & Sedoc. 2019. Conceptor Debiasing of Word Representations Evaluated on WEAT. Proceedings of the First Workshop on Gender Bias in Natural Language Processing.

Ma, Xu, Wang, Nallapati, & Xiang. 2019. Universal Text Representation from BERT: An Empirical Study.

Manzini, Lim, Tsvetkov, Black. 2019. Black Is to Criminal as Caucasian Is to Police: Detecting and Removing Multiclass Bias in Word Embeddings. Proceedings of the 2019 Conference of the North.

Nikhil Garg, Londa Schiebinger, Dan Jurafsky, and James Zou. 2018. Word embeddings quantify 100 years of gender and ethnic stereotypes. Proc. of the National Academy of Sciences, 115(16).

Tianlu Wang, Jieyu Zhao, Mark Yatskar, Ryan Cotterell, Vicente Ordonez, and Kai-Wei Chang. 2019. Gender bias in contextualized word embeddings. North American Chapter of the Association for Computational Linguistics (NAACL).

Appendix A. Analysis of Attention Heads

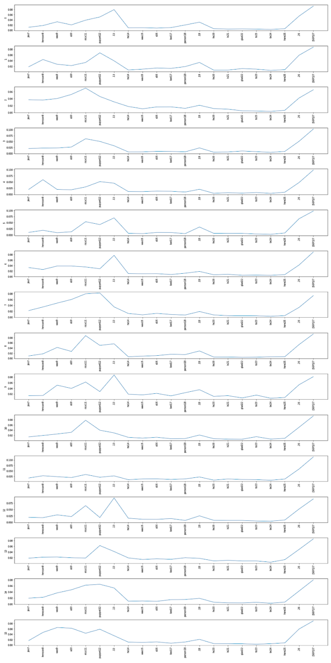


Fig. 6. A representation of mean activation, by head.

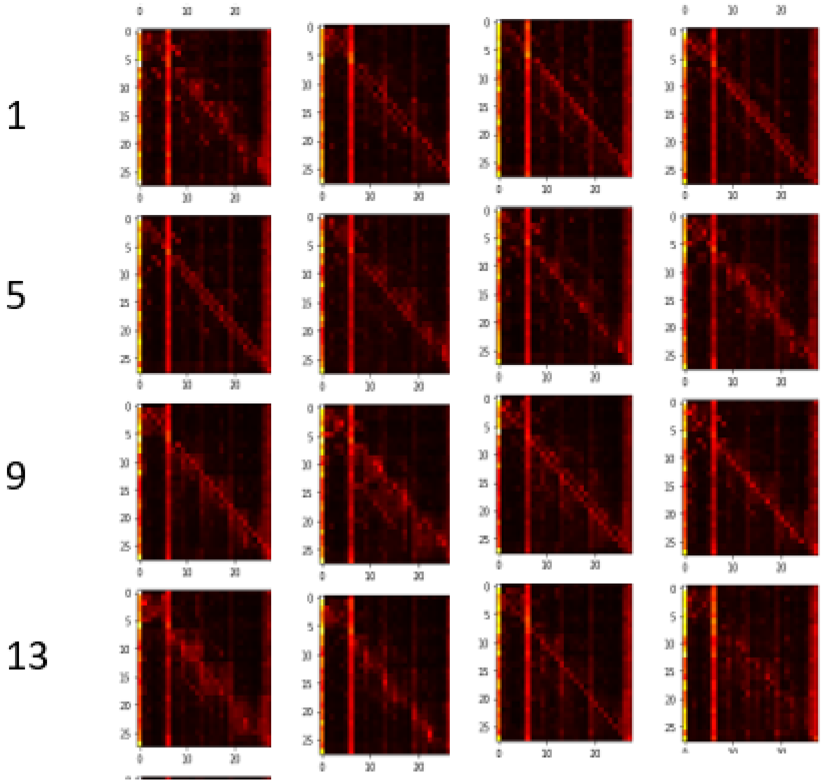


Fig. 7. A depiction of the activation matrix, by head.